Scheduling Dual-Arm Cluster Tools With Multiple Wafer Types and Residency Time Constraints

Jipeng Wang, *Student Member, IEEE*, Hesuan Hu, *Senior Member, IEEE*, Chunrong Pan, *Senior Member, IEEE*, Yuan Zhou, and Liang Li

***Abstract*— Accompanying the unceasing progress of integrated circuit manufacturing technology, the mainstream production mode of current semiconductor wafer fabrication is featured with multi-variety, small batch, and individual customization, which poses a huge challenge to the scheduling of cluster tools with single-wafer-type fabrication. Concurrent processing multiple wafer types in cluster tools, as a novel production pattern, has drawn increasing attention from industry to academia, whereas the corresponding research remains insufficient. This paper investigates the scheduling problems of dual-arm cluster tools with multiple wafer types and residency time constraints. To pursue an easy-to-implement cyclic operation under diverse flow patterns, we develop a novel robot activity strategy called multiplex swap sequence. In the light of the virtual module technology, the workloads that stem from bottleneck process steps and asymmetrical process configuration are balanced satisfactorily. Moreover, several sufficient and necessary conditions with closed-form expressions are obtained for checking the system's schedulability. Finally, efficient algorithms with polynomial complexity are developed to find the periodic scheduling, and its practicability and availability are demonstrated by the offered illustrative examples.**

***Index Terms*—Cluster tools, multiple wafer types, scheduling, semiconductor manufacturing, wafer fabrication.**

NOMENCLATURE

$\mathbb{N}^+$	$\{1, 2, \ldots\}$.
$\mathbb{N}_n$	$\{1, 2, \ldots, n\}$.
Ω_n	$\mathbb{N}_n \cup \{0\}$.
$\mathbb{N}_{k-\gamma}^{k}$	$\mathbb{N}_k \backslash \mathbb{N}_{k-\gamma}$.
$\mathbb{S}$	$\{\lambda \in \mathbb{N}_k, \mu \in \mathbb{N}_{n_\lambda}, \varphi_{\lambda\mu \mathrm{U}} < \varphi_{\mathrm{Lmax}}\}$.
$\mathbb{R}$	$\{\lambda \in \mathbb{N}_k, \mu \in \mathbb{N}_{n_\lambda}, \varphi_{\lambda\mu \mathrm{U}} \geq \varphi_{\mathrm{Lmax}}\}$.
$\mathbb{G}_\lambda^0$	$\{1, 2, \ldots, \mu_\lambda^1 - 1\}$.
$\mathbb{G}_\lambda^\delta$	$\{\mu_\lambda^\delta, \mu_\lambda^\delta + 1, \ldots, \mu_\lambda^{\delta+1} - 1\}$.
$\mathbb{G}_\lambda^\varepsilon$	$\{\mu_\lambda^\varepsilon, \mu_\lambda^\varepsilon + 1, \ldots, n_\lambda\}$.
k	The number of the total wafer types.
γ	The number of the total shared types.
ε	The number of the total shared process steps.
κ_λ	The number of wafer type λ in the fabrication order.
κ_λ^v	The number of the added virtual wafer of type λ.
$m_{\lambda\mu}$	The number of parallel processing modules (PMs) of wafer type λ at process step μ, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
n_λ	The number of the total process steps of wafer type λ, $\lambda \in \mathbb{N}_k$.
ψ	Cycle time of the system.
$\psi_{\sigma R}$	Cycle time of the robot in the σth subcycle, $\sigma \in \mathbb{N}_\gamma$.
$\psi_{\lambda\mu}$	Cycle time of wafer type λ at process step μ, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\alpha_{\lambda\mu}$	Wafer processing time of type λ at process step μ, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\beta_{\lambda\mu}$	The longest permissive time a wafer of type λ can stay at process step μ after being processed.
ρ	The time needed for the robot to unload/load a wafer from/into the PM or loadlock (LL).
θ	The time needed for the robot to move between different modules.
π	The time needed for performing the simple swap operation at a process step.
$\tau_{\lambda\mu}$	Wafer sojourn time of type λ at process step μ, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\omega_{\lambda\mu}^P$	Robot waiting time at process step μ of wafer type λ before the swap operation, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\omega_{\lambda\mu}^R$	Robot waiting time at process step μ of wafer type λ during the swap operation, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\zeta_{\lambda\mu}$	The coefficient with respect to the shared behaviors, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
$\xi_{\lambda\mu}$	The coefficient with respect to the internal virtual modules, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$.
η_λ	The coefficient with respect to the external virtual modules, $\lambda \in \mathbb{N}_k$.
R_{opt}	The optimum quantitative proportion of the entire wafer types.

Manuscript received February 10, 2020; accepted March 22, 2020. This work was supported in part by the National Natural Science Foundation of China (71361014, 61973242, 61573265, 51665018) and the Major Fundamental Research Program of the Natural Science Foundation of Shaanxi Province (2017ZDJC-34). Recommended by Associate Editor MengChu Zhou. *(Corresponding author: Hesuan Hu and Chunrong Pan.)*

Citation: J. P. Wang, H. S. Hu, C. R. Pan, Y. Zhou, and L. Li, "Scheduling dual-arm cluster tools with multiple wafer types and residency time constraints," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 776–789, May 2020.

J. P. Wang is with the School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: jp.wang@stu.xidian.edu.cn).

H. Hu is with the School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China, and also with the School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: huhesuan@gmail.com).

C. R. Pan is with the School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China (e-mail: crpan@jxust.edu.cn).

Y. Zhou is with the School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: y.zhou@ntu.edu.sg).

L. Li is with the School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China, and also with the Department of Information Engineering, Electrical Engineering and Applied Mathematics, University of Salerno, Fisciano 84084, Italy (e-mail: liangli@stu.xidian.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2020.1003150

I. Introduction

In semiconductor manufacturing industry, cluster tools are the most fundamental equipment. For the sake of a higher tool utilization rate and a higher production yield rate, the majority of wafer manufacturers apply cluster tools in various wafer fabrication processes. Particularly in the cutting-edge fabs, about 70%~80% of the wafer fabrication procedures, such as wafer cleaning, etching, and deposition, are performed in cluster tools [1]. Owing to the interior space restriction, a typical radical-type cluster tool is generally composed of several single-wafer processing modules (PMs, also called processing chambers), a wafer transfer module (TM, also called wafer-handling robot), and two loadlocks (LLs). All these modules are mechanically connected together and manipulated by the industrial computer. PMs process wafers, TM executes wafer loading, unloading, and transportation among different modules, and LLs load/unload wafer cassettes. Depending on the robot structure, cluster tools are classified into the single- and dual-arm cluster tool (see Fig. 1). Furthermore, a couple of individual cluster tools can be interconnected through buffer modules in linear or treelike topology architecture to form a multi-cluster tool [2]–[4].

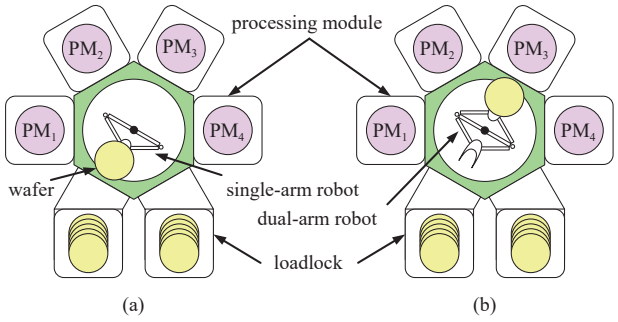


Fig. 1. Cluster tools with four PMs. (a) A single-arm cluster tool; (b) A dual-arm cluster tool.

Cluster tools are a sort of integrated equipment that provides a fully automated job shop manufacturing environment with vacuum conditions. More importantly, it is a big-ticket item ranging in price from hundreds of thousands to millions of dollars per tool [5], which consumes huge investments and production costs. Thus, it is of great necessity to operate cluster tools efficiently for reducing production costs. Great works regarding the modeling, analysis, and performance evaluation of cluster tools have been conducted [2], [3], [6]–[9]. Meanwhile, it is usually taken for granted that the nature of the operation of cluster tools is to schedule the robot activities due to its domination in PM activities [8]. An indisputable fact that the time taken for wafer processing in PMs is much longer than that for robot activities has been acknowledged by the most of the past research with regard to cluster tools. Thus under most circumstances, cluster tools operate in the process-bound region, where the robot has idle time and the wafer processing in PMs determines the system cycle time. In addition, the backward and swap sequences show their efficiencies in operating single- and dual-arm cluster tools respectively [6]. Nevertheless, the complex restrictiveness, stemming from the equipment architecture, wafer recipes, and temporal logic, poses significant challenges to the scheduling of cluster tools. Over the past decades, researchers conducted extensive prominent studies with respect to scheduling cluster tools under diverse instances, including wafer residency time constraints (WRTCs) [10]–[15], activity time variation [16]–[20], wafer revisiting processes [21]–[27], PM failures [28], [29], chamber cleaning operation [1], [30]–[32], and multi-cluster tools [33]–[42], which are thoroughly reviewed in [43]. Moreover, the latest advancements can be found in [44], [45]. Note that all the aforementioned works are dedicated to the scheduling and control problems of single-wafer-type fabrication.

With the rapid improvements of integrated circuit (IC) manufacturing technology, the wafer size has increased up to 300 mm even 450 mm; besides, main leading-edge IC fabricators have stepped into 14 nm technology node, thus facilitating the reduction of wafer lot size. To respond to the customer order as quickly as possible, wafer manufacturers have to frequently switch wafer lots with a higher rapidity. If switching wafer lots with the off-line way, there is little prospect of minimizing the system's makespan due to the repeated equipment shutdown for computing schedules and adjusting PMs [46]. If, instead, switching wafer lots with the on-line way, it does not need to close and adjust cluster tools frequently [47]. However, it remains an inefficient methodology due to the multiple occurrences of transient processes. For the transient process scheduling, its intractability in the solvability and enforceability is aggravated by the frequent state evolution, even though there has been some corresponding studies [48]–[53]. Meanwhile, the idleness of some PMs can hardly be avoided within each switching process, and this reduces the tool's efficiency. In addition, despite the fact that existing research has made efforts so as to switch two different wafer lots in real-time, none of them provides a feasible integrated resolution for the scenarios with more than two lots. In brief, the intractable issues outlined in above can scarcely be solved via the resolutions for cluster tools with single-wafer-type fabrication. As a consequence, to tackle with these issues, novel approaches need to be explored.

Recently, the leading fabs attempt to process several wafer types in a cluster tool concurrently. Such a novel wafer fabrication mode shows promising prospects on handling the new scheduling challenges caused by the shrinking wafer lot size. By considering that two different wafer types are processed in the cluster tool concurrently, the corresponding scheduling problems have been discussed in [54], [55]. Lee *et al.* [54] assume that there is no PM shared by the two wafer types, and then present efficient heuristic scheduling rules to determine the optimal robot activity sequence. A mixed integer programming model is established for minimizing the cycle time, when the developed conditions do not hold. Later, Lee *et al.* [55] extend their previous results in [54] to the tool environment where a single PM is shared by two different wafer types. By combining both the alternating and the conventional robot activity sequence, a heuristic algorithm is proposed for operating the robot in the general cycle. Ko *et al.*

[56] investigate the scheduling problem for the dual-arm cluster tool that concurrently processes multiple wafer types with identical flow patterns but different process times at each process step. This is a more common practice of concurrent processing in actual fabs. Looking back to the existing works [54]–[56], the commonality is that none of them takes WRTCs into consideration when addressing the concurrent processing problems. In our earlier work [46], we investigate the concurrent processing of multiple wafer types in a single-arm cluster tools. We place little restriction on the number of the entire wafer types and the flow pattern of each wafer type, i.e., the concurrent processing of multiple wafer types conducted in [46] includes, but is not limited to, the instances that have been discussed in [54], [55]. Meanwhile, we take WRTCs as the core condition in [46]. As known, compared with single-arm cluster tools, dual-arm ones have higher productivity [5]–[8], [43]. Notwithstanding this, corresponding issues have not been extended to dual-arm cluster tools platform. However, the approach for single-arm cluster tools presented in [46] cannot be applied in dual-arm ones due to diverse differences including in the structure and resource behaviors. Thus, it remains an open problem and deserves exploring novel methodologies.

In this paper, we focus on the scheduling problems of dual-arm cluster tools with multiple wafer types. To improve the effectiveness and adaptability in engineering practice, we integratedly take several constraints into consideration including WRTCs, diverse flow patterns, and quantitative proportion. Considering them together in this paper, we make threefold contributions: 1) present a novel robot operation strategy that reveals effectiveness for the simplicity of its enforcement and the flexibility of tackling with intricate combination of flow patterns; 2) derive several necessary and sufficient conditions with closed-form expressions for verifying the system's schedulability; and 3) develop efficient algorithms with polynomial complexity for computing the optimal cyclic schedule when the system is schedulable.

The rest of this paper is organized as follows. Section II introduces necessary backgrounds on the concurrent processing of multiple wafer types in dual-arm cluster tools. Section III proposes a novel robot activity sequence to enforce cyclic operation, and then analyzes temporal properties and systematic workloads. Section IV discusses the schedulability criteria and presents efficient algorithms for finding the cyclic scheduling. Section V offers several examples to demonstrate the proposed methodology's efficiency and its enforcement method. Finally, Section VI summarizes this paper and suggests possible extensions.

II. Problem Preliminaries

For the dual-arm cluster tool scheduling, there are alternative operation strategies, including dynamic (or real-time) and static (or off-line) dispatching rules of the robot activities. The dynamic strategy usually results in aperiodic operations, unpredictable robot activities, complex computation, and intractability in scheduling implementation, even though the corresponding optimizing issues can be heuristically performed via computer simulation. With the static strategy, a schedule can be obtained in advance based on the operation objectives and requirements. In other words, such a strategy pursues cyclic operation, i.e., the robot repeats identical work cycle and so does each PM. Cyclic operation outperforms the aperiodic one thanks to various merits, such as reduced scheduling complexity, predictable behavior, less work-in-progress inventory, improved throughput, and steady (or periodical) timing patterns. Thus, this paper is devoted to the cyclic scheduling of dual-arm cluster tools with multiple wafer types and WRTCs.

A. Problem Definition

The problem investigated in this paper is the concurrent processing of multiple wafer types in dual-arm cluster tools. For the basic positioning, each wafer type has individual recipe and quantity. Meanwhile, WRTCs within PMs are considered. In practice, it remains possible that some wafer types possess several identical process steps that demand the same operation and conditions, i.e., processing technology, processing time, and WRTCs. To maximize the utilization of module resources, tool engineers would assign the same PMs, namely the shared PMs, to different wafer types for performing these identical process steps. For simplicity, we designate the wafer type with (resp., without) shared PMs as the shared (resp., non-shared) type, the process step with (resp., without) shared PMs as the shared (resp., non-shared) process step as well. Then, for the problem defined in this paper, we can provide a unified description, i.e., a production order is composed of k wafer types, γ ones of which are the shared types, where $k \in \mathbb{N}^+\backslash\{1\}$, $\mathbb{N}^+ = \{1, 2, \ldots\}$, and $\gamma \in \mathbb{N}_k = \{1, 2, \ldots, k\}$. Let κ_λ denote the quantity of wafer type λ. Then, the total quantity of a production order with k wafer types is $\sum_{\lambda=1}^{k}\kappa_\lambda$, $\lambda \in \mathbb{N}_k$.

In general, the wafer technological process is described as the wafer flow pattern (WFP). Excluding revisiting processes, a general single type wafer flow pattern is defined as $(m_1, m_2, \ldots, m_n)$ [11], where m_i denotes the number of parallel PMs at process step i, $i \in \mathbb{N}_n$, and n is the number of process steps. Corresponding to the single type WFP, the definition for the multi-type one should have the capability to describe the recipe of each wafer type. In addition, the PM sharing behavior should be characterized in an appropriate way. Based on the specification introduced in [46], the WFP for multiple wafer types can be defined as $\{(m_{11}, m_{12}, \ldots, m_{1n_1}), (m_{21}, m_{22}, \ldots, m_{2n_2}), \ldots, (m_{(k-\gamma)1}, m_{(k-\gamma)2}, \ldots, m_{(k-\gamma)n_{k-\gamma}}), (m_{(k-\gamma+1)1}, m_{(k-\gamma+1)2}, \ldots, [m_{(k-\gamma+1)\mu^1_{k-\gamma+1}}], \ldots, [m_{(k-\gamma+1)\mu^2_{k-\gamma+1}}], \ldots, [m_{(k-\gamma+1)\mu^\varepsilon_{k-\gamma+1}}], \ldots, m_{(k-\gamma+1)n_{k-\gamma+1}}), (m_{(k-\gamma+2)1}, m_{(k-\gamma+2)2}, \ldots, [m_{(k-\gamma+2)\mu^1_{k-\gamma+2}}], \ldots, [m_{(k-\gamma+2)\mu^2_{k-\gamma+2}}], \ldots, [m_{(k-\gamma+2)\mu^\varepsilon_{k-\gamma+2}}], \ldots, m_{(k-\gamma+2)n_{k-\gamma+2}}), \ldots, (m_{k1}, m_{k2}, \ldots, [m_{k\mu^1_k}], \ldots, [m_{k\mu^2_k}], \ldots, [m_{k\mu^\varepsilon_{k-\gamma+1}}], \ldots, m_{kn_k})\}$, where n_λ ($\lambda \in \mathbb{N}_k$) is the number of the total process steps of wafer type λ, $m_{\lambda\mu}$ ($\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$) is the number of parallel PMs of wafer type λ at the non-shared process step μ, and $m_{\lambda\mu^\delta_\lambda}$ ($\lambda \in \mathbb{N}_k\backslash\mathbb{N}_{k-\gamma}$, $\delta \in \mathbb{N}_\varepsilon$) is the number of parallel PMs of wafer type λ at the shared process step μ^δ_λ, i.e., the δth process step with shared PMs. In this definition, we adopt the square bracket "[ ]" to mark the shared process

steps. In addition, ε denotes the number of the entire shared process steps. Thus, $\varepsilon \leq \min\{n_{k-\gamma+1}, n_{k-\gamma+2}, \ldots, n_k\}$, i.e., $\varepsilon \in \{1, 2, \ldots, \min\{n_{k-\gamma+1}, n_{k-\gamma+2}, \ldots, n_k\}\}$.

B. Activity Description

To schedule a cluster tool, it encompasses two properties, i.e., the performing activity sequence and time in such manufacturing systems. As pointed out in Section I, the swap operation is an efficient strategy for the dual-arm cluster tool scheduling. Based on the swap strategy at process step μ of wafer type λ, the robot will perform the following activity sequence: move to process step $\mu \rightarrow$ wait for a completed wafer at process step $\mu \rightarrow$ swap-with-wait at process step μ. Let $\Upsilon_{\lambda\mu}$ represent such events executed by the robot. Further, let $\Upsilon_{\lambda i} \bigoplus \Upsilon_{\lambda j}$ denote that after executing $\Upsilon_{\lambda i}$, events $\Upsilon_{\lambda(i+1)}$, $\Upsilon_{\lambda(i+2)}$, …, $\Upsilon_{\lambda j}$ will be successively followed. Then, we have $\Upsilon_{\lambda i} \bigoplus \Upsilon_{\lambda j} = \Upsilon_{\lambda i} \rightarrow \Upsilon_{\lambda(i+1)} \rightarrow \cdots \rightarrow \Upsilon_{\lambda j}$, where $\lambda \in \mathbb{N}_k$, $i, j \in \mathbb{N}^+$ and $i < j$.

Time is taken for both PMs and robot activities. Let $\alpha_{\lambda\mu}$ denote the process time of wafer type λ at process step μ. Due to the residency time constraints, the wafer of type λ must be unloaded within $\beta_{\lambda\mu}$ time units at most after its completion in the PM at process step μ. This implies that the wafer sojourn time $\tau_{\lambda\mu}$ must fall into the time interval $[\alpha_{\lambda\mu}, \alpha_{\lambda\mu} + \beta_{\lambda\mu}]$, i.e., $\tau_{\lambda\mu} \in [\alpha_{\lambda\mu}, \alpha_{\lambda\mu} + \beta_{\lambda\mu}]$. As the fact that the loadlock has no processing function, we thus have $\alpha_{\lambda 0} = 0$ and $\tau_{\lambda 0} \in [0, +\infty)$. As done in [11]–[14], [21], [56], we assume that: 1) the time taken for loading a wafer is the same as that for unloading, denoted as ρ, and 2) the time taken for the robot to move among any two different modules is identical, denoted as θ. Such assumption is universally accepted since the variability of these activities is marginal, and their operation times are much shorter than the wafer processing time in practice. In addition, when the robot rotates to the process step μ of wafer type λ for performing the swap operation, the wafer may remain being processed in the PM. To make the schedule feasible, the robot may need to wait some time units before or during the swap operations, which are denoted as $\omega^P_{\lambda\mu}$ and $\omega^R_{\lambda\mu}$, respectively. Assume that it takes π time units for performing a simple swap operation at a process step. Then, the time taken for performing $\Upsilon_{\lambda\mu}$ is $\theta + \omega^P_{\lambda\mu} + \pi + \omega^R_{\lambda\mu}$.

III. Scheduling Analysis

In this section, we introduce a novel robot activity strategy for achieving the cyclic fabrication of multiple wafer types in cluster tools without difficulty. On the basis of the analysis of the process step in temporal and workload properties, we present approaches to compute corresponding cycle times and to balance systematic workloads.

A. Multiplex Swap Sequence

It follows from the discussion with respect to the concurrent processing of multiple wafer types in the single-arm cluster tool that the production processes of each wafer type are mutually independent [46], likewise in the dual-arm cluster tool. This provides a holistic perspective combining the local and global cycle to examine the fabrication processes of different wafer types. As described in the multi-type WFP, the raw wafers include two parts. One is the $k-\gamma$ non-shared types, while another is the γ shared ones. Assume that the shared types enter the tool before the non-shared ones. Then, the wafer type release sequence is [the shared types] $\rightarrow$ [wafer type $k-\gamma$] $\rightarrow$ [wafer type $k-\gamma-1$] $\rightarrow \cdots \rightarrow$ [wafer type 1]. However, according to the structural characteristics revealed by the multi-type WFP, we can safely draw a conclusion that the shared types have shown a parallelized structure mutually from the global perspective. Therefore, based on the first-in first-out principle, the wafer type release sequence in the global cycle composed of γ subcycles is : $\{[\text{the shared types}]_k \rightarrow [\text{wafer type } k-\gamma] \rightarrow [\text{wafer type } k-\gamma-1] \rightarrow \cdots \rightarrow [\text{wafer type 1}]\}_1 \rightarrow \{[\text{the shared types}]_{k-1} \rightarrow [\text{wafer type } k-\gamma] \rightarrow [\text{wafer type } k-\gamma-1] \rightarrow \cdots \rightarrow [\text{wafer type 1}]\}_2 \rightarrow \cdots \rightarrow \{[\text{the shared types}]_{k-\gamma+1} \rightarrow [\text{wafer type } k-\gamma] \rightarrow [\text{wafer type } k-\gamma-1] \rightarrow \cdots \rightarrow [\text{wafer type 1}]\}_\gamma$, where the subscript σ, $\sigma \in \mathbb{N}_\gamma$, in "[the shared types]$_\sigma$" indicates that the wafer being processed in the final shared process step is the σth type. Thus, the system will evolve to its initial state after γ subcycles.

In an arbitrary global cycle, the non-shared types are processed γ times with an identical sequence, while the shared ones require to be dynamically executed in each subcycle with different sequences. Let $\mathrm{PS}_{\lambda\mu}$, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$, denote the μth process step of wafer type λ. Specially, the LLs can be treated as a process step, denoted as $\mathrm{PS}_{\lambda 0}$. Let $\Gamma_\lambda^{i\rightarrow j}$ represent the process step sequence for wafer type λ from the ith to jth process steps, i.e., $\Gamma_\lambda^{i\rightarrow j} = \mathrm{PS}_{\lambda i} \rightarrow \mathrm{PS}_{\lambda(i+1)} \rightarrow \cdots \rightarrow \mathrm{PS}_{\lambda j}$. For the non-shared types, according to the swap-with-wait operation, the robot will execute process steps sequentially in each subcycle as follows: $\Gamma_1^{0\rightarrow n_1} \rightarrow \Gamma_2^{0\rightarrow n_2} \rightarrow \cdots \rightarrow \Gamma_{k-\lambda}^{0\rightarrow n_{k-\gamma}}$. As for the shared types, their fabrication mechanisms are more complicated. Due to the parallelized structure and identical manufacturing parameters at the shared process steps, each shared PM conducts a different wafer type and configures a permutation in the light of the first-in first-out rule. Thus, in the σth subcycle, wafers being processed from the first to the final shared process steps are $\mathrm{W}_{\vartheta(\sigma,1)}$, $\mathrm{W}_{\vartheta(\sigma,2)}$, …, $\mathrm{W}_{\vartheta(\sigma,\delta)}$, …, $\mathrm{W}_{\vartheta(\sigma,\varepsilon)}$, respectively, where W_λ indicates the wafer of the λth type. Note that $\vartheta_{(\sigma,\delta)}$ is equal to $k - \langle \varepsilon + \sigma - \delta | \gamma \rangle + 1$, where $\langle \varepsilon + \sigma - \delta | \gamma \rangle$ is equal to γ when $\varepsilon + \sigma - \delta$ is divisible by γ; otherwise, $\langle \varepsilon + \sigma - \delta | \gamma \rangle$ is the remainder of $\varepsilon + \sigma - \delta$ by dividing γ. In two unbiased adjacent shared process steps, the process step chain composed of the identical wafer type satisfying the latter instance will be chosen to execute. In the σth subcycle, thus, the robot will execute the process steps of the shared types sequentially as follows: $\Gamma_{\vartheta(\sigma,0)}^{\mu^0_{\vartheta(\sigma,0)}\rightarrow\mu^1_{\vartheta(\sigma,0)}-1} \rightarrow \Gamma_{\vartheta(\sigma,1)}^{\mu^1_{\vartheta(\sigma,1)}\rightarrow\mu^2_{\vartheta(\sigma,1)}-1} \rightarrow \cdots \rightarrow \Gamma_{\vartheta(\sigma,\varepsilon)}^{\mu^\varepsilon_{\vartheta(\sigma,\varepsilon)}\rightarrow n_{\vartheta(\sigma,\varepsilon)}}$.

By this point, the robot operation strategy with respect to the non-shared and shared types is demonstrated clearly within the above discussions. Then, based on the standpoint that the wafer fabrications of the non-shared and shared types (as a unit) are mutually independent, the process steps of these two parts can be concatenated and executed with the swap-

with-wait operation to construct the multiplex swap sequence (MSS), i.e., $[\Gamma_1^{0\to n_1} \to \Gamma_2^{0\to n_2} \to \cdots \to \Gamma_{k-\lambda}^{0\to n_{k-\gamma}} \to \Gamma_{\vartheta(1,0)}^{0\to \mu^1_{\vartheta(1,0)}-1} \to \Gamma_{\vartheta(1,1)}^{\mu^1_{\vartheta(1,1)}\to \mu^2_{\vartheta(1,1)}-1} \to \cdots \to \Gamma_{\vartheta(1,\varepsilon)}^{\mu^\varepsilon_{\vartheta(1,\varepsilon)}\to n_{\vartheta(1,\varepsilon)}}]_1 \to [\Gamma_1^{0\to n_1} \to \Gamma_2^{0\to n_2} \to \cdots \to \Gamma_{k-\lambda}^{0\to n_{k-\gamma}} \to \Gamma_{\vartheta(2,0)}^{0\to \mu^1_{\vartheta(2,0)}-1} \to \Gamma_{\vartheta(2,1)}^{\mu^1_{\vartheta(2,1)}\to \mu^2_{\vartheta(2,1)}-1} \to \cdots \to \Gamma_{\vartheta(2,\varepsilon)}^{\mu^\varepsilon_{\vartheta(2,\varepsilon)}\to n_{\vartheta(2,\varepsilon)}}]_2 \to \cdots \to [\Gamma_1^{0\to n_1} \to \Gamma_2^{0\to n_2} \to \cdots \to \Gamma_{k-\lambda}^{0\to n_{k-\gamma}} \to \Gamma_{\vartheta(\gamma,0)}^{0\to \mu^1_{\vartheta(\gamma,0)}-1} \to \Gamma_{\vartheta(\gamma,1)}^{\mu^1_{\vartheta(\gamma,1)}\to \mu^2_{\vartheta(\gamma,1)}-1} \to \cdots \to \Gamma_{\vartheta(\gamma,\varepsilon)}^{\mu^\varepsilon_{\vartheta(\gamma,\varepsilon)}\to n_{\vartheta(\gamma,\varepsilon)}}]_\gamma$.

B. Temporal Analysis

According to the results in [11], [14], we can easily draw the conclusion that the time needed for completing a wafer of type λ at process step μ is $\pi + \omega^R_{\lambda\mu} + \tau_{\lambda\mu}$. In consideration of the concurrent mechanism featured by two aspects, i.e., the parallel PMs and the parallelized structure that are formulated by the shared and non-shared PMs of the shared wafer types, we characterize such specification by introducing two coefficients: $m_{\lambda\mu}$ and $\zeta_{\lambda\mu}$, where $m_{\lambda\mu}$ is the number of parallel PMs of $\mathrm{PS}_{\lambda\mu}$; and $\zeta_{\lambda\mu} = 1$ if $\lambda \in \mathbb{N}_{k-\gamma}$ or $\mu = \mu^\delta_\lambda$; otherwise, $\zeta_{\lambda\mu} = \gamma$. Thus, the production cycle time of $\mathrm{PS}_{\lambda\mu}$, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$, is

$$\psi_{\lambda\mu} = \frac{\pi + \omega^R_{\lambda\mu} + \tau_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}}. \tag{1}$$

In accordance with the MSS, the process step sequence that will be performed is easily identified. Given the preliminary assumption along this paper, the elementary operation during each process step is performed with swap-with-wait policy. Then in the σth subcycle, the executed process step sequence is $[\Gamma_1^{0\to n_1} \to \Gamma_2^{0\to n_2} \to \cdots \to \Gamma_{k-\lambda}^{0\to n_{k-\gamma}} \to \Gamma_{\vartheta(\sigma,0)}^{0\to \mu^1_{\vartheta(\sigma,0)}-1} \to \Gamma_{\vartheta(\sigma,1)}^{\mu^1_{\vartheta(\sigma,1)}\to \mu^2_{\vartheta(\sigma,1)}-1} \to \cdots \to \Gamma_{\vartheta(\sigma,\varepsilon)}^{\mu^\varepsilon_{\vartheta(\sigma,\varepsilon)}\to n_{\vartheta(\sigma,\varepsilon)}}]_\sigma$. Thus, the robot activity sequence is $[\Upsilon_{10} \bigoplus \Upsilon_{1n_1} \to \Upsilon_{20} \bigoplus \Upsilon_{2n_2} \to \cdots \to \Upsilon_{(k-\lambda)0} \bigoplus \Upsilon_{(k-\lambda)n_{k-\lambda}} \to \Upsilon_{\vartheta(\sigma,0)\mu^0_{\vartheta(\sigma,0)}} \bigoplus \Upsilon_{\vartheta(\sigma,0)(\mu^1_{\vartheta(\sigma,0)}-1)} \to \Upsilon_{\vartheta(\sigma,1)\mu^1_{\vartheta(\sigma,1)}} \bigoplus \Upsilon_{\vartheta(\sigma,1)(\mu^2_{\vartheta(\sigma,1)}-1)} \to \cdots \to \Upsilon_{\vartheta(\sigma,\varepsilon)\mu^\varepsilon_{\vartheta(\sigma,\varepsilon)}} \bigoplus \Upsilon_{\vartheta(\sigma,\varepsilon)n_{\vartheta(\sigma,\varepsilon)}}]$. Consequently, the robot cycle time in the σth subcycle is

$$\begin{aligned}\psi_{\sigma R} = {} & [\sum_{\lambda=1}^{k-\gamma} n_\lambda + k - \gamma + \sum_{\delta=0}^{\varepsilon-1}(\mu^{\delta+1}_{\vartheta(\sigma,\delta+1)} - \mu^\delta_{\vartheta(\sigma,\delta)}) + n_{\vartheta(\sigma,\varepsilon)} \\ & - \mu^\varepsilon_{\vartheta(\sigma,\varepsilon)} + 1](\pi+\theta) + \sum_{\delta=0}^{\varepsilon-1}\sum_{\mu=\mu^\delta_{\vartheta(\sigma,\delta)}}^{\mu^{\delta+1}_{\vartheta(\sigma,\delta)}-1}(\omega^P_{\vartheta(\sigma,\delta)\mu} + \omega^R_{\vartheta(\sigma,\delta)\mu}) \\ & + \sum_{\mu=\mu^\varepsilon_{\vartheta(\sigma,\delta)}}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega^P_{\vartheta(\sigma,\varepsilon)\mu} + \omega^R_{\vartheta(\sigma,\varepsilon)\mu}) + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega^P_{\lambda\mu} + \omega^R_{\lambda\mu}).\end{aligned} \tag{2}$$

With the MSS, based on the previous analysis, when the system arrives at the steady state, the wafer fabrication process in dual-arm cluster tools presents a cyclic serial one. Hence, the cycle time for completing a wafer at each step and the robot cycle time are identical. This is unified as the system production cycle time, denoted as ψ. Thus, $\forall\ \lambda \in \mathbb{N}_k$, $\sigma \in \mathbb{N}_\gamma$, we have

$$\psi = \psi_{\lambda 1} = \psi_{\lambda 2} = \cdots = \psi_{\lambda n_\lambda} \tag{3}$$

$$\psi = \psi_{\sigma R}. \tag{4}$$

Equations (3) and (4) present the basics for scheduling cluster tools with multiple types of wafers within the steady state, as well as the properties with respect to the temporal relationship and productivity between each pair of modules under cyclic operations. For the wafer fabrication within each process step in the steady state, as revealed by (3) and (4), one should and must keep the same production rhythm, which can be guaranteed via alternative approaches including adjusting the wafer fabrication starting time and the robot waiting time. As for the robot, due to the anisotropic distribution of the process step chain among two adjacent shared process steps, its activity cycle usually emerges as a dynamic one resulting in unstable work cycle and error propagation of the wafer fabrication in PMs. To obtain a stable robot activity cycle, in the bottom line, for $\forall\ \delta \in \Omega_{\varepsilon-1} = \mathbb{N}_{\varepsilon-1} \cup \{0\}$, $i, j \in \mathbb{N}^k_{k-\gamma} = \mathbb{N}_k \backslash \mathbb{N}_{k-\gamma}$, and $i \neq j$, we must ensure

$$\begin{aligned}&(\mu_i^{\delta+1} - \mu_i^\delta - 1)(\pi+\theta) + \sum_{\mu=\mu_i^\delta+1}^{\mu_i^{\delta+1}-1}(\omega^P_{i\mu} + \omega^R_{i\mu}) \\ &\quad = (\mu_j^{\delta+1} - \mu_j^\delta - 1)(\pi+\theta) + \sum_{\mu=\mu_j^\delta+1}^{\mu_j^{\delta+1}-1}(\omega^P_{j\mu} + \omega^R_{j\mu})\end{aligned} \tag{5}$$

$$\begin{aligned}&(n_i - \mu_i^\varepsilon - 1)(\pi+\theta) + \sum_{\mu=\mu_i^\varepsilon+1}^{n_i}(\omega^P_{i\mu} + \omega^R_{i\mu}) \\ &\quad = (n_j - \mu_j^\varepsilon - 1)(\pi+\theta) + \sum_{\mu=\mu_j^\varepsilon+1}^{n_j}(\omega^P_{j\mu} + \omega^R_{j\mu}).\end{aligned} \tag{6}$$

According to the assumptions with respect to the shared process steps, i.e., $\forall\ \delta \in \Omega_{\varepsilon-1}$, $i, j \in \mathbb{N}^k_{k-\gamma}$, and $i \neq j$, $\exists\ \omega^P_{i\mu_i^\delta} = \omega^P_{j\mu_i^\delta}$ and $\omega^R_{i\mu_i^\delta} = \omega^R_{j\mu_i^\delta}$, we thus have

$$\begin{aligned}&(\mu_i^{\delta+1} - \mu_i^\delta)(\pi+\theta) + \sum_{\mu=\mu_i^\delta}^{\mu_i^{\delta+1}-1}(\omega^P_{i\mu} + \omega^R_{i\mu}) \\ &\quad = (\mu_i^{\delta+1} - \mu_i^\delta)(\pi+\theta) + \sum_{\mu=\mu_j^\delta}^{\mu_j^{\delta+1}-1}(\omega^P_{j\mu} + \omega^R_{j\mu})\end{aligned} \tag{7}$$

$$\begin{aligned}&(n_i - \mu_i^\varepsilon + 1)(\pi+\theta) + \sum_{\mu=\mu_j^\varepsilon}^{n_j}(\omega^P_{i\mu} + \omega^R_{i\mu}) \\ &\quad = (n_j - \mu_j^\varepsilon + 1)(\pi+\theta) + \sum_{\mu=\mu_j^\varepsilon}^{n_j}(\omega^P_{j\mu} + \omega^R_{j\mu}).\end{aligned} \tag{8}$$

Consequently, for the tools' operation in the steady state, (2) should be updated to

$$\psi_{\sigma R} = [\sum_{\lambda=1}^{k-\gamma} n_\lambda + k - \gamma + n_{\vartheta(\sigma,\varepsilon)} + 1](\pi+\theta) + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega_{\lambda\mu}^P + \omega_{\lambda\mu}^R) + \sum_{\mu=0}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega_{\vartheta(\sigma,\varepsilon)\mu}^P + \omega_{\vartheta(\sigma,\varepsilon)\mu}^R) = \psi_{R_c} + \psi_{\sigma R_w}. \quad (9)$$

where

$$\psi_{R_c} = (\sum_{\lambda=1}^{k-\gamma} n_\lambda + k - \gamma + n_{\vartheta(\sigma,\varepsilon)} + 1)(\pi+\theta) \quad (10)$$

$$\psi_{\sigma R_w} = \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega_{\lambda\mu}^P + \omega_{\lambda\mu}^R) + \sum_{\mu=0}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega_{\vartheta(\sigma,\varepsilon)\mu}^P + \omega_{\vartheta(\sigma,\varepsilon)\mu}^R). \quad (11)$$

C. Workload Analysis and Balance

Due to the incongruous parametric distribution and the fierce resource contention that occur in the shared PMs, the system workload imbalance probably deteriorates to a severe level such that additional measures are needed, apart from the robot waiting time adjustment that is widely applied in [13], [14], [17]–[18], [22]–[25]. To tackle with these issues and realize stable cyclic operations, we adopt the virtual module technology that is introduced in our earlier work [46]. In the internal layer, we import $[\max_{i\in\mathbb{N}_{k-\gamma}^k}\{\mu_i^1 - \mu_i^0\} - (\mu_\lambda^1 - \mu_\lambda^0)](\pi + \theta)$ (resp., $[\max_{i\in\mathbb{N}_{k-\gamma}^k}\{n_i - \mu_i^\varepsilon\} - (n_\lambda - \mu_\lambda^\varepsilon)](\pi + \theta)$) virtual modules as well as the regulating waiting $\omega_\lambda^{V_b}$ ($\omega_\lambda^{V_a}$) before (resp., after) the first (resp., final) shared process step. Since there is no buffer between neighboring PMs, we have to allocate virtual robot operation to the process step $\mu_\lambda^\sigma - 1$. In the external layer, we set $\eta_\lambda - 1$ virtual PM groups for each wafer type. In particular, the shared types are regarded as an individual group, i.e., $\eta_{k-\gamma+1} = \eta_{k-\gamma+2} = \ldots = \eta_k = \eta_s$. Then, (1) for computing the production cycle time of $\mathrm{PS}_{\lambda\mu}$ should be updated as

$$\psi_{\lambda\mu} = \frac{\xi_{\lambda\mu}(\pi+\theta) + \pi + \omega_{\lambda\mu}^R + \tau_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda} \quad (12)$$

where

$$\xi_{\lambda\mu} = \begin{cases} \max\limits_{i\in\mathbb{N}_{k-\gamma}^k}\{\mu_i^\delta - \mu_i^{\delta-1}\} - (\mu_\lambda^\delta - \mu_\lambda^{\delta-1}), \\ \qquad \text{if } \mu = \mu_\lambda^\delta - 1, \lambda \in \mathbb{N}_{k-\gamma}^k, \delta \in \mathbb{N}_\varepsilon \backslash \{1\} \\ 0, \qquad \text{otherwise} \end{cases} \quad (13)$$

$$\eta_\lambda = \left\lceil \frac{\max\limits_{\mu\in\mathbb{N}_{n_\lambda}}\left\{\frac{\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}}\right\}}{\psi_{R_c}} - \frac{1}{2} \right\rceil \quad (14)$$

and $\lceil x \rceil$ indicates the smallest integer that is not less than x. Then, the lower and upper bounds of the production cycle time of $\mathrm{PS}_{\lambda\mu}$ respectively are

$$\psi_{\lambda\mu\mathrm{L}} = \frac{\xi_{\lambda\mu}(\pi+\theta) + \pi + \omega_{\lambda\mu}^R + \alpha_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda} \quad (15)$$

$$\psi_{\lambda\mu\mathrm{U}} = \frac{\xi_{\lambda\mu}(\pi+\theta) + \pi + \omega_{\lambda\mu}^R + \alpha_{\lambda\mu} + \beta_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda}. \quad (16)$$

Equation (12) characterizes not only the production cycle time but also the workload range of each process step under a specific schedule. Because the specification defined by (12) includes the robot waiting time, we call such a workload as the scheduled workload that has a close relation with the robot cycle time. Their algebraic relationships can be maximally characterized by the coefficient η_λ. Consequently, the optimum quantitative proportion of the wafer types can be determined as follows:

$$R_{\mathrm{opt}} = \frac{1}{\eta_1} : \frac{1}{\eta_2} : \cdots : \frac{1}{\eta_{k-\gamma}} : \frac{1}{\gamma\eta_{k-\gamma+1}} : \frac{1}{\gamma\eta_{k-\gamma+2}} : \cdots : \frac{1}{\gamma\eta_k}. \quad (17)$$

After configuring the virtual modules, the robot activity cycle can be converted into a reductive specification as follows:

$$\psi_{\sigma R} = (\sum_{\lambda=1}^{k-\gamma} n_\lambda + \sum_{\delta=0}^{\varepsilon-1}\max_{i\in\mathbb{N}_{k-\gamma}^k}\{\mu_i^{\delta+1} - \mu_i^\delta\} + \max_{j\in\mathbb{N}_{k-\gamma}^k}\{n_j - \mu_j^\varepsilon\} + k - \gamma + 1)(\pi+\theta) + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega_{\lambda\mu}^P + \omega_{\lambda\mu}^R) + \sum_{\mu=0}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega_{\vartheta(\sigma,\varepsilon)\mu}^P + \omega_{\vartheta(\sigma,\varepsilon)\mu}^R) + \omega_{\vartheta(\sigma,\varepsilon)}^{V_b} + \omega_{\vartheta(\sigma,\varepsilon)}^{V_a} = \psi_{R_c} + \psi_{\sigma R_w} \quad (18)$$

where

$$\psi_{R_c} = (\sum_{\lambda=1}^{k-\gamma} n_\lambda + \sum_{\delta=0}^{\varepsilon-1}\max_{i\in\mathbb{N}_{k-\gamma}^k}\{\mu_i^{\delta+1} - \mu_i^\delta\} + \max_{j\in\mathbb{N}_{k-\gamma}^k}\{n_j - \mu_j^\varepsilon\} + k - \gamma + 1)(\pi+\theta) \quad (19)$$

$$\psi_{\sigma R_w} = \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega_{\lambda\mu}^P + \omega_{\lambda\mu}^R) + \sum_{\mu=0}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega_{\vartheta(\sigma,\varepsilon)\mu}^P + \omega_{\vartheta(\sigma,\varepsilon)\mu}^R) + \omega_{\vartheta(\sigma,\varepsilon)}^{V_b} + \omega_{\vartheta(\sigma,\varepsilon)}^{V_a}. \quad (20)$$

It should be noticed that ψ_{R_c} is a predeterminable constant related to the inherent characteristics of the system including the WFP, processing times, and robot activity times, whereas $\psi_{\sigma R_w}$ shall be determined by the specific schedule pursued in this paper. By using virtual module technology, we can alleviate the process step bottleneck in both internal and external layers. Moreover, when the system runs in the steady state, it can be inferred from (3), (4) and (12) that the wafer sojourn time at $\mathrm{PS}_{\lambda\mu}$ is

$$\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi+\theta) + \pi + \omega_{\lambda\mu}^R]. \quad (21)$$

IV. Schedulability and Scheduling

By this point, we have discussed the robot operation strategy and the corresponding temporal properties. The sequential work is to verify the feasibility of the target schedule that satisfies (3), (4) and WRTCs, i.e., $\psi_{\lambda\mu} \in [\psi_{\lambda\mu\mathrm{L}}, \psi_{\lambda\mu\mathrm{U}}]$, simultaneously. To be specific, the problem converts into how to clarify the variational mechanism of the

production cycle time based on the limited manufacturing information that can be known in advance. Although such a solution cannot be derived directly, we can handle it by analyzing the distribution characteristic of the natural workload of $\mathrm{PS}_{\lambda\mu}$. Removing the robot waiting time $\omega^R_{\lambda\mu}$ from (12), we can obtain $\mathrm{PS}_{\lambda\mu}$'s natural workload $\varphi_{\lambda\mu}$ and its lower and upper bounds as follows

$$\varphi_{\lambda\mu} = \frac{\xi_{\lambda\mu}(\pi+\theta)+\pi+\tau_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda} \tag{22}$$

$$\varphi_{\lambda\mu\mathrm{L}} = \frac{\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda} \tag{23}$$

$$\varphi_{\lambda\mu\mathrm{U}} = \frac{\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}+\beta_{\lambda\mu}}{m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda}. \tag{24}$$

Let $\varphi_{\mathrm{Lmax}} = \max\{\varphi_{\lambda\mu\mathrm{L}},\ \lambda \in \mathbb{N}_k,\ \mu \in \mathbb{N}_{n_\lambda}\}$ and $\varphi_{\mathrm{Umin}} = \min\{\varphi_{\lambda\mu\mathrm{U}},\ \lambda \in \mathbb{N}_k,\ \mu \in \mathbb{N}_{n_\lambda}\}$. Next, we investigate the systematic schedulability under the diverse distribution characteristic in manufacturing parameters.

A. Situation With $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$

When $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$, (3) and (4) are easy to guarantee without excessive intervention by the robot. The remaining problem is to deconstruct the wafer residency mechanism in PMs as well as its relationship with the systematic schedulability. Based on the value of ψ_{R_c}, as shown in Fig. 2, the manufacturing parameter distribution can be divided into three cases: 1) $\psi_{R_c} \le \varphi_{\mathrm{Lmax}}$, 2) $\varphi_{\mathrm{Lmax}} < \psi_{R_c} \le \varphi_{\mathrm{Umin}}$, and 3) $\psi_{R_c} > \varphi_{\mathrm{Umin}}$. Then, we have the following theorem.

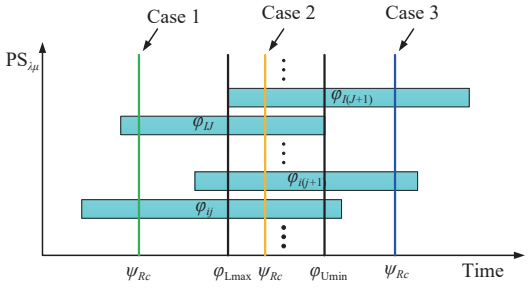


Fig. 2. Illustration for Theorem 1.

Theorem 1: For the time-constrained dual-arm cluster tool with $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$, if $\psi_{R_c} \le \varphi_{\mathrm{Umin}}$, then, the system is schedulable; otherwise, it is not schedulable.

Proof: Case 1: Let $\omega^P_{10} = \varphi_{\mathrm{Lmax}} - \psi_{R_c}$, and assign zero to the rest of robot waiting times. Then, we have $\psi_{\sigma R_w} = \varphi_{\mathrm{Lmax}} - \psi_{R_c}$ and the robot cycle time $\psi_{\sigma R} = \psi_{R_c} + \psi_{\sigma R_w} = \varphi_{\mathrm{Lmax}}$. Thus, the time taken for completing a wafer in each process step is φ_{Lmax} in the steady state based on (3) and (4). Therefore, $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \varphi_{\mathrm{Lmax}}$, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$, $\sigma \in \mathbb{N}_\gamma$. Then, from (21), we have $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi+\theta)+\pi+\omega^R_{\lambda\mu}] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{\mathrm{Lmax}} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \ge m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}]/(m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda) - [\xi_{\lambda\mu}(\pi+\theta)+\pi] = \alpha_{\lambda\mu}$, and $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi+\theta)+\pi\ \omega^R_{\lambda\mu}] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{R_c} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \le m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\mathrm{Umin}} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \le m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}+\beta_{\lambda\mu}]/(m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda) - [\xi_{\lambda\mu}(\pi+\theta)+\pi] = \alpha_{\lambda\mu}+\beta_{\lambda\mu}$, i.e., $\tau_{\lambda\mu} \in [\alpha_{\lambda\mu},\ \alpha_{\lambda\mu}+\beta_{\lambda\mu}]$. In other words, there is enough time for completing a wafer in each process step; besides, the WRTC of each process step is satisfied. Thus, the system is schedulable for Case 1.

Case 2: For this case, let all the robot waiting times be zero, such that $\psi_{\sigma R_w} = 0$ and the robot cycle time $\psi_{\sigma R} = \psi_{R_c} + \psi_{\sigma R_w} = \psi_{R_c}$. According to (3) and (4), in the steady state, the time taken for completing a wafer in each process step is forced to be ψ_{R_c}, i.e., $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \psi_{R_c}$, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$, $\sigma \in \mathbb{N}_\gamma$. Then, from (21), we have $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi+\theta)+\pi+\omega^R_{\lambda\mu}] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{R_c} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] > m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{\mathrm{Lmax}} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \ge m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}]/(m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda) - [\xi_{\lambda\mu}(\pi+\theta)+\pi] = \alpha_{\lambda\mu}$, and $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi+\theta)+\pi+\omega^R_{\lambda\mu}] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{R_c} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \le m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\mathrm{Umin}} - [\xi_{\lambda\mu}(\pi+\theta)+\pi] \le m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\xi_{\lambda\mu}(\pi+\theta)+\pi+\alpha_{\lambda\mu}+\beta_{\lambda\mu}]/(m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda) - [\xi_{\lambda\mu}(\pi+\theta)+\pi] = \alpha_{\lambda\mu}+\beta_{\lambda\mu}$, i.e., $\tau_{\lambda\mu} \in [\alpha_{\lambda\mu},\ \alpha_{\lambda\mu}+\beta_{\lambda\mu}]$. This means that the WRTC of each process step is satisfied. Thus, the system is schedulable for Case 2.

Case 3: It is clear that the production cycle time $\psi > \psi_{R_c}$ if the system is schedulable. According to (12), (18), and (21), we can easily deduce that $\psi_{\sigma R_w} \ge (\psi - \varphi_{\mathrm{Umin}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda$. Then, $\psi_{\sigma R} = \psi_{R_c} + \psi_{\sigma R_w} \ge \psi_{R_c} + (\psi - \varphi_{\mathrm{Umin}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda > \psi_{R_c} + (\psi - \psi_{R_c})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda = \psi + (m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda - 1)(\psi - \psi_{R_c}) > \psi$. Thus, there always exists $\psi_{\sigma R} > \psi$ violating (4) no matter what schedules we adopt. This implies that the system is not schedulable for Case 3. ■

Summarizing the aforementioned discussions with respect to the situation with $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$, we present Algorithm 1 to find the periodic steady state schedule for time-constrained dual-arm cluster tools with multiple wafer types.

Algorithm 1 Scheduling time-constrained dual-arm cluster tools with multiple wafer types for $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$

Input: WFP, $\alpha_{\lambda\mu}$, $\beta_{\lambda\mu}$, ρ, θ
Output: ψ, R_{opt}, $\omega^P_{\lambda\mu}$, $\omega^R_{\lambda\mu}$, $\omega^{V_b}_\lambda$, $\omega^{V_a}_\lambda$
1 **Calculating** φ_{Lmax}, φ_{Umin}, η_λ, R_{opt}, and ψ_{R_c};
2 **if** $\psi_{R_c} \le \varphi_{\mathrm{Lmax}}$ **then**
3 $\psi \leftarrow \varphi_{\mathrm{Lmax}}$;
4 $\omega^P_{\lambda\mu} \leftarrow \psi - \psi_{R_c}$, if $\lambda = 1$ and $\mu = 0$;
5 Otherwise, $\omega^P_{\lambda\mu} \leftarrow 0$;
6 $\omega^R_{\lambda\mu} \leftarrow 0$, $\lambda \in \mathbb{N}_k$, $\mu \in \Omega_{n_\lambda}$;
7 $\omega^{V_b}_\lambda \leftarrow 0$, $\omega^{V_a}_\lambda \leftarrow 0$, $\lambda \in \mathbb{N}^k_{k-\gamma}$;
8 **else**
9 **if** $\psi_{R_c} \le \varphi_{\mathrm{Umin}}$ **then**
10 $\psi \leftarrow \psi_{R_c}$;
11 $\omega^P_{\lambda\mu} \leftarrow 0$, $\omega^R_{\lambda\mu} \leftarrow 0$, $\lambda \in \mathbb{N}_k$, $\mu \in \Omega_{n_\lambda}$;
12 $\omega^{V_b}_\lambda \leftarrow 0$, $\omega^{V_a}_\lambda \leftarrow 0$, $\lambda \in \mathbb{N}^k_{k-\gamma}$;
13 **else**
14 the system is not schedulable
15 **end**
16 **end**

B. Situation With $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] = \emptyset$

As revealed in the above statements, if the systematic

manufacturing parameter distribution is characterized with $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}}, \varphi_{\lambda\mu\mathrm{U}}] \neq \emptyset$ and $\psi_{R_c} \in (-\infty, \varphi_{\mathrm{Umin}}]$, we can find the feasible schedule with no effort since there is no more interference when regulating the robot waiting times. However, there are still possibilities of $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}}, \varphi_{\lambda\mu\mathrm{U}}] = \emptyset$, i.e., $\varphi_{\mathrm{Lmax}} > \varphi_{\mathrm{Umin}}$. Let $\mathbb{S} = \{\lambda \in \mathbb{N}_k, \mu \in \mathbb{N}_{n_\lambda}, \varphi_{\lambda\mu\mathrm{U}} < \varphi_{\mathrm{Lmax}}\}$, $\mathbb{R} = \{\lambda \in \mathbb{N}_k, \mu \in \mathbb{N}_{n_\lambda}, \varphi_{\lambda\mu\mathrm{U}} \geq \varphi_{\mathrm{Lmax}}\}$, $\mathbb{G}_\lambda^0 = \{1, 2, \ldots, \mu_\lambda^1 - 1\}$, $\mathbb{G}_\lambda^\delta = \{\mu_\lambda^\delta, \mu_\lambda^\delta + 1, \ldots, \mu_\lambda^{\delta+1} - 1\}$, and $\mathbb{G}_\lambda^\varepsilon = \{\mu_\lambda^\varepsilon, \mu_\lambda^\varepsilon + 1, \ldots, n_\lambda\}$, $\delta \in \mathbb{N}_{\varepsilon-1}$, $\lambda \in \mathbb{N}_{k-\gamma}^k$. Let $\Psi = \varphi_{\mathrm{Lmax}} - [\sum_{\lambda\in\mathbb{N}_{k-\gamma}\cap\mathbb{S}}\sum_{\mu\in\mathbb{N}_{n_\lambda}\cap\mathbb{S}} + \sum_{\delta=0}^{\varepsilon}\max_{\lambda\in\mathbb{N}_{k-\gamma}^k\cap\mathbb{S}}\sum_{\mu\in\mathbb{G}_\lambda^\delta\cap\mathbb{S}}](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda$. Likewise, based on the value of ψ_{R_c}, as shown in Fig. 3, the manufacturing parameter distribution can be divided into three cases: 1) $\psi_{R_c} \leq \Psi$, 2) $\Psi < \psi_{R_c} \leq \varphi_{\mathrm{Lmax}}$, and 3) $\psi_{R_c} > \varphi_{\mathrm{Lmax}}$. Then, we have the following theorem.

Theorem 2: For the time-constrained dual-arm cluster tool with $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] = \emptyset$, if $\psi_{R_c} \leq \Psi$, then, the system is schedulable; otherwise, it is not schedulable.

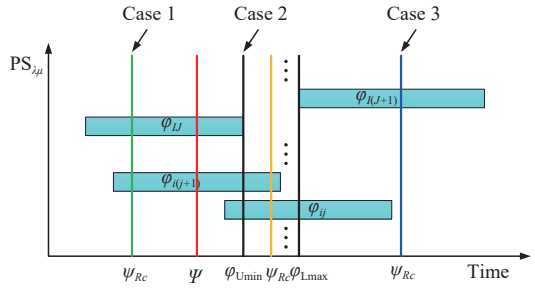


Fig. 3. Illustration for Theorem 2.

Proof: Case 1: For the sake of simplicity, let $\sum_1$ and $\sum_2$ denote $\sum_{\lambda\in\mathbb{N}_{k-\gamma}\cap\mathbb{S}}\sum_{\mu\in\mathbb{N}_{n_\lambda}\cap\mathbb{S}}$ and $\sum_{\delta=0}^{\varepsilon}\max_{\lambda\in\mathbb{N}_{k-\gamma}^k\cap\mathbb{S}}\sum_{\mu\in\mathbb{G}_\lambda^\delta\cap\mathbb{S}}$, respectively. Due to $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}}, \varphi_{\lambda\mu\mathrm{U}}] = \emptyset$, we cannot find a feasible schedule such that $\psi \in (-\infty, \varphi_{\mathrm{Lmax}})$. Thus, the production cycle time ψ should belong to $[\varphi_{\mathrm{Lmax}}, +\infty)$ if the system is schedulable. Let $\omega_{\lambda\mu}^R = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\mathrm{Lmax}} - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \alpha_{\lambda\mu} + \beta_{\lambda\mu}]$, $\lambda \in \mathbb{N}_k \cap \mathbb{S}$, $\mu \in \mathbb{N}_{n_\lambda} \cap \mathbb{S}$, and $\omega_{\lambda\mu}^R = 0$, $\lambda \in \mathbb{N}_k \cap \mathbb{R}$, $\mu \in \mathbb{N}_{n_\lambda} \cap \mathbb{R}$. Then, we have $\psi_{\sigma R_w} = \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}(\omega_{\lambda\mu}^P + \omega_{\lambda\mu}^R) + \sum_{\mu=0}^{n_{\vartheta(\sigma,\varepsilon)}}(\omega_{\vartheta(\sigma,\varepsilon)\mu}^P + \omega_{\vartheta(\sigma,\varepsilon)\mu}^R) + \omega_{\vartheta(\sigma,\varepsilon)}^{V_b} + \omega_{\vartheta(\sigma,\varepsilon)}^{V_a} = [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})\, m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^P$. It follows from $\psi_{R_c} \leq \varphi_{\mathrm{Lmax}} - [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda$ that $\sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^P = \psi_{\sigma R_w} - [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda = \varphi_{\mathrm{Lmax}} - [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda - \psi_{R_c} \geq 0$. Therefore, we can find a common solution φ_{Lmax} for $\psi_{\lambda\mu}$ and $\psi_{\sigma R}$, i.e., $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \varphi_{\mathrm{Lmax}}$. Based on Case 1 of Theorem 1, it is clear that each process step belonging to $\mathbb{R}$ satisfies the residency time constraints. For the robot activities in $\mathbb{S}$, we have $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega_{\lambda\mu}^R] > m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi_{\lambda\mu\mathrm{S}} - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega_{\lambda\mu}^R] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\xi_{\lambda\mu}(\pi + \theta) + \pi + \alpha_{\lambda\mu} + \omega_{\lambda\mu}^R]/(m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda) - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega_{\lambda\mu}^R] = \alpha_{\lambda\mu}$, and $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega_{\lambda\mu}^R] \leq m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\mathrm{Lmax}} - [\xi_{\lambda\mu}(\pi + \theta) + \pi + (\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\mathrm{Lmax}} - (\varphi_{\mathrm{Lmax}}m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda - \alpha_{\lambda\mu} - \beta_{\lambda\mu}) = \alpha_{\lambda\mu} + \beta_{\lambda\mu}$, i.e., $\tau_{\lambda\mu} \in (\alpha_{\lambda\mu}, \alpha_{\lambda\mu} + \beta_{\lambda\mu}]$. Thus, the WRTC of each process step belonging to $\mathbb{S}$ is also satisfied. In consequence, the system can be scheduled for Case 1.

Case 2: It is clear that the production cycle time $\psi \geq \varphi_{\mathrm{Lmax}}$ if the feasible schedule can be found. From $\Psi < \psi_{R_c} \leq \varphi_{\mathrm{Lmax}}$, we have $[\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda > \varphi_{\mathrm{Lmax}} - \psi_{R_c}$. Then, $\psi_{\sigma R} = \psi_{R_c} + \psi_{\sigma R_w} = \psi_{R_c} + [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})\, m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^P \geq \psi_{R_c} + [\sum_1 + \sum_2](\psi - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda = \psi_{R_c} + [\sum_1 + \sum_2](\psi - \varphi_{\mathrm{Lmax}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda > \psi_{R_c} + [\sum_1 + \sum_2](\psi - \varphi_{\mathrm{Lmax}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + \varphi_{\mathrm{Lmax}} - \psi_{R_c} = [\sum_1 + \sum_2](\psi - \varphi_{\mathrm{Lmax}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + \varphi_{\mathrm{Lmax}} > \psi$. Thus, the schedule is infeasible, i.e., the system is not schedulable under the above two conditions for Case 2.

Case 3: It follows from $\psi_{R_c} > \varphi_{\mathrm{Lmax}} > \varphi_{\mathrm{Umin}}$ that $\psi > \psi_{R_c}$ if the system is schedulable. Then, $\psi_{\sigma R} = \psi_{R_c} + \psi_{\sigma R_w} = \psi_{R_c} + [\sum_1 + \sum_2](\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}})\, m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^P \geq \psi_{R_c} + [\sum_1 + \sum_2](\psi - \varphi_{\lambda\mu\mathrm{U}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda > \psi_{R_c} + (\psi - \varphi_{\mathrm{Umin}})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda > \psi_{R_c} + (\psi - \psi_{R_c})m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda = \psi + (m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda - 1)(\psi - \psi_{R_c}) > \psi$. Thus, any feasible schedule matching the steady state requirements cannot be found, i.e., the system is not schedulable for Case 3. ■

Summarizing the aforementioned discussions with respect to the situation with $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] = \emptyset$, we have Algorithm 2.

Algorithm 2 Scheduling time-constrained dual-arm cluster tools with multiple wafer types for $\bigcap_{\lambda\in\mathbb{N}_k,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}},\varphi_{\lambda\mu\mathrm{U}}] = \emptyset$

Input: WFP, $\alpha_{\lambda\mu}$, $\beta_{\lambda\mu}$, ρ, θ
Output: ψ, R_{opt}, $\omega_{\lambda\mu}^P$, $\omega_{\lambda\mu}^R$, $\omega_\lambda^{V_b}$, $\omega_\lambda^{V_a}$
Calculating φ_{Lmax}, φ_{Umin}, η_λ, R_{opt}, and ψ_{R_c};
if $\psi_{R_c} \leq \varphi_{\mathrm{Lmax}}$ **then**
  if $\psi_{R_c} \leq \Psi$ **then**
    $\psi \leftarrow \varphi_{\mathrm{Lmax}}$;
    $\omega_{\lambda\mu}^R \leftarrow 0$, $\lambda \in \mathbb{N}_k \cap \mathbb{R}$, $\mu \in \mathbb{N}_{n_\lambda} \cap \mathbb{R}$;
    $\omega_{\lambda\mu}^R \leftarrow m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda[\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}}]$, $\lambda \in \mathbb{N}_k \cap \mathbb{S}$, $\mu \in \mathbb{N}_{n_\lambda} \cap \mathbb{S}$;
    $\Delta_{\lambda\delta} \leftarrow \max_{i\in\mathbb{N}_{k-\gamma}^k\cap\mathbb{S}}\sum_{j\in\mathbb{G}_i^\delta\cap\mathbb{S}}[\varphi_{\mathrm{Lmax}} - \varphi_{ij\mathrm{U}}] - \sum_{\mu\in\mathbb{G}_\lambda^\delta\cap\mathbb{S}}[\varphi_{\mathrm{Lmax}} - \varphi_{\lambda\mu\mathrm{U}}]$, $\lambda \in \mathbb{N}_{k-\gamma}^k$, $\delta \in \mathbb{N}_{\varepsilon-1}$;
    $\omega_{\lambda(\mu_\lambda^\delta - 1)}^P \leftarrow m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\Delta_{\lambda\delta}$, $\lambda \in \mathbb{N}_{k-\gamma}^k$, $\delta \in \mathbb{N}_{\varepsilon-1}$;
    $\omega_{\lambda\mu}^P \leftarrow 0$, $\lambda \in \mathbb{N}_{k-\gamma}^k$, $\mu \in \mathbb{G}_\lambda^\delta \backslash \{\mu_\lambda^\delta - 1\}$, $\delta \in \mathbb{N}_{\varepsilon-1}$;
    $\omega_{\lambda\mu}^P \leftarrow 0$, $\lambda \in \mathbb{N}_{k-\gamma}^k$, $\mu \in \mathbb{G}_\lambda^0 \cup \mathbb{G}_\lambda^\varepsilon \cap \{0\}$;
    $\omega_\lambda^{V_b} \leftarrow \max\sum_{j=0}^{\mu_i^1-1}\omega_{ij}^R - \sum_{\mu=0}^{\mu_\lambda^1-1}\omega_{\lambda\mu}^R$, $i, \lambda \in \mathbb{N}_{k-\gamma}^k$;
    $\omega_\lambda^{V_a} \leftarrow \max\sum_{j=\mu_i^\varepsilon}^{n_i}\omega_{ij}^R - \sum_{\mu=\mu_\lambda^\varepsilon}^{n_\lambda}\omega_{\lambda\mu}^R$, $i, \lambda \in \mathbb{N}_{k-\gamma}^k$;
    $\omega_{\lambda\mu}^P \leftarrow 0$, $\lambda \in \mathbb{N}_{k-\gamma} \backslash \{1\}$, $\mu \in \Omega_{n_\lambda}$;
    $\omega_{1\mu}^P \leftarrow 0$, $\mu \in \mathbb{N}_{n_1}$;
    $\omega_{10}^P \leftarrow \varphi_{\mathrm{Lmax}} - \psi_{R_c} - [\omega_i^{V_b} + \sum_{\lambda=2}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^P + \sum_{\mu=1}^{n_1}\omega_{1\mu}^P + \sum_{\lambda=1}^{k-\gamma}\sum_{\mu=0}^{n_\lambda}\omega_{\lambda\mu}^R + \sum_{\mu=0}^{n_i}(\omega_{i\mu}^P + \omega_{i\mu}^R) + \omega_i^{V_a}]$, $\forall\, i \in \mathbb{N}_{k-\gamma}^k$;
  else
    the system is not schedulable
  end
else
  the system is not schedulable
end

C. Discussion

In Section III, we propose a novel robot operation strategy and the corresponding workload balancing methods. They provide a comprehensive approach to cope with the concurrent processing of multiple wafer types in dual-arm cluster tools. Combining with the schedulability criteria provided in this section, we can summarize the procedure to find the feasible cyclic schedule for dual-arm cluster tools with multiple wafer types and residency time constraints. As for the performance of the proposed algorithm, we have the following theorem.

Theorem 3: If the mixing ratio of the production order coincides with R_{opt}, the schedule obtained by Algorithms 1 and 2 is optimal in terms of the cycle time.

Proof: Assume that the lower natural workload of $\text{PS}_{\lambda\mu}$, $\lambda \in \mathbb{N}_k$, $\mu \in \mathbb{N}_{n_\lambda}$, $\varphi_{\lambda\mu\text{L}}$ is equal to φ_{Lmax}. For the situations covered by Case 1 of Theorems 1 and 2, if the system is scheduled such that $\psi < \varphi_{\text{Lmax}}$, then, according to (21), for the $\text{PS}_{\lambda\mu}$, we have $\tau_{\lambda\mu} = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\psi - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega^R_{\lambda\mu}] < m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\text{Lmax}} - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega^R_{\lambda\mu}] = m_{\lambda\mu}\zeta_{\lambda\mu}\eta_\lambda\varphi_{\lambda\mu\text{L}} - [\xi_{\lambda\mu}(\pi + \theta) + \pi + \omega^R_{\lambda\mu}] = \alpha_{\lambda\mu} - \omega^R_{\lambda\mu} < \alpha_{\lambda\mu}$. Thus, such schedule is infeasible. For the situation defined by the Case 2 of Theorem 1, if the system is scheduled such that $\psi < \varphi_{R_c}$, then, according to (3) and (4), we have $\psi_{\gamma R} = \psi < \psi_{R_c}$ that contradicts with the definition of the robot activity cycle. Therefore, such schedule is infeasible. In conclusion, the system driven by the obtained schedule can reach the lower bound of the production cycle time, i.e., the minimal cycle time. By considering the presented schedulability conditions that are constructed in terms of the quantitative proportion (see (12), (17), and (21)), the entire wafer types will be completed simultaneously within the identical processing cycle. Thus, the system can keep working throughout the whole processes with the minimal cycle time under the supposed conditions. ■

As stated above, the tool system can operate in the lower bound continually on the basis of the assumption that the mixing ratio of a specific production order is equal to R_{opt}. However, the mixing ratio is inconsistent with R_{opt} in many scenarios. In such cases, although the obtained schedule may no longer be the optimal one, it is still efficient due to the simple computation and implementation. In addition, we can adopt the virtual wafer technology, as implemented in [13], [14], to coordinate the robot activities. More specifically, we need to configure the mixing ratio in accordance with R_{opt} by adding virtual wafers minimally. After configuring, thus, we have $(\kappa_1 + \kappa_1^v) : (\kappa_2 + \kappa_2^v) : \cdots : (\kappa_k + \kappa_k^v) = R_{\text{opt}}$, where κ_λ^v denotes the number of the added virtual wafers of type λ. Given the WFP of a production order, one needs to calculate the cycle time of n_λ process steps for the λ-th type of wafers. Thus, the computational complexity of the presented algorithm is $O(n_\lambda \times K)$. However, by considering that $n_\lambda \leq 6$ and K is limited in practice, the presented approach is efficient and applicable to industrial cases.

V. Examples

In this section, we give several examples to demonstrate the effectiveness of the proposed algorithm as well as its implementation approach.

A. Illustrative Examples

Example 1: The processing paths of the first, second, and third wafer types are [LL → PM_1 → PM_2 → LL], [LL → PM_3 → PM_4 → LL], and [LL → PM_5 → PM_6 → LL], respectively. The processing parameters are as follows: κ_1 = 240 pcs, α_{11} = 286 s, β_{11} = 26 s, α_{12} = 264 s, β_{12} = 30 s; κ_2 = 180 pcs, α_{21} = 128 s, β_{21} = 15 s, α_{22} = 110 s, β_{22} = 28 s; κ_3 = 600 pcs, α_{31} = 120 s, β_{31} = 24 s, α_{32} = 133 s, β_{32} = 17 s; ρ = 8 s, θ = 3 s, π = 19 s.

For this example, the WFP is {(1, 1), (1, 1), (1, 1)}. Then, $\psi_{R_c} = (\sum_{\lambda=1}^{3} n_\lambda + 3)(\pi + \theta)$ = 198 s, η_1 = 2, η_2 = 3, and η_3 = 1. Thus, we have $\varphi_{11\text{L}}$ = 194 s, $\varphi_{11\text{U}}$ = 207 s, $\varphi_{12\text{L}}$ = 204 s, $\varphi_{12\text{U}}$ = 213 s, $\varphi_{21\text{L}}$ = 196 s, $\varphi_{21\text{U}}$ = 206 s, $\varphi_{22\text{L}}$ = 202 s, $\varphi_{22\text{U}}$ = 211 s, $\varphi_{31\text{L}}$ = 204 s, $\varphi_{31\text{U}}$ = 218 s, $\varphi_{32\text{L}}$ = 185 s, and $\varphi_{32\text{U}}$ = 206 s. Then, $\bigcap_{\lambda\in\mathbb{N}_3,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\text{L}}, \varphi_{\lambda\mu\text{U}}] \neq \emptyset$, and $\psi_{R_c} < \varphi_{\text{Lmax}}$ = 204 s. Hence, the conditions given by Case 1 of Theorem 1 are satisfied. According to Algorithm 1, a feasible cyclic schedule can be obtained by setting $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \varphi_{\text{Lmax}}$ = 204 s, $\omega_{10}^P = \varphi_{\text{Lmax}} - \psi_{R_c}$ = 6 s, and the rest of robot waiting times are equal to zeros. The optimal quantity ratio of wafer types is R_{opt} = 3 : 2 : 6. Thus, the quantities of the virtual wafers added necessarily are as follows: κ_1^v = 60 pcs, κ_2^v = 20 pcs, and κ_3^v = 0 pcs. Fig. 4 depicts the Gantt chart of the obtained schedule.

Example 2: The processing paths of the first and second wafer types are [LL → PM_1 → PM_2 (PM_3) → PM_4 → PM_6 → LL] and [LL → PM_1 → PM_5 → PM_6 → LL], respectively. The processing parameters are as follows: κ_1 = 480 pcs, α_{11} = 145 s, β_{11} = 16 s, α_{12} = 145 s, β_{12} = 16 s, α_{13} = 145 s, β_{13} = 16 s, α_{14} = 145 s, β_{14} = 16 s; κ_2 = 200 pcs, α_{21} = 348 s, β_{21} = 24 s, α_{22} = 326 s, β_{22} = 28 s, α_{23} = 145 s, β_{23} = 16 s; ρ = 10 s, θ = 2 s, π = 22 s.

For this example, the WFP is {([1], 2, 1, [1]), ([1], 1, [1])}. Then, ψ_{R_c} = (4 + 1) × (22 + 2) = 120 s. Thus, we have $\varphi_{11\text{L}}$ = 118 s, $\varphi_{11\text{U}}$ = 130 s, $\varphi_{12\text{L}}$ = 116 s, $\varphi_{12\text{U}}$ = 123 s, $\varphi_{13\text{L}}$ = 117 s, $\varphi_{13\text{U}}$ = 130 s, $\varphi_{14\text{L}}$ = 103 s, $\varphi_{14\text{U}}$ = 128 s, $\varphi_{21\text{L}}$ = 118 s, $\varphi_{21\text{U}}$ = 130 s, $\varphi_{22\text{L}}$ = 109 s, $\varphi_{22\text{U}}$ = 124 s, $\varphi_{23\text{L}}$ = 103 s, and $\varphi_{23\text{U}}$ = 128 s. Then, $\bigcap_{\lambda\in\mathbb{N}_2,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\text{L}}, \varphi_{\lambda\mu\text{U}}] \neq \emptyset$, and φ_{Lmax} (118 s) < ψ_{R_c} (120 s) < φ_{Umin} (123 s). Hence, the conditions given in Case 2 of Theorem 1 are satisfied. According to Algorithm 1, a feasible cyclic schedule can be obtained by setting $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \psi_{R_c}$ = 120 s, and all the robot waiting times are equal to zeros. The optimal quantity ratio of wafer types is R_{opt} = 1 : 1. Thus, the quantities of the virtual wafers added necessarily are as follows: κ_1^v = 20 pcs and κ_2^v = 50 pcs. Fig. 5 depicts the Gantt chart of the obtained schedule.

Example 3: The processing paths of the first, second, and third wafer types are [LL → PM_1(PM_2) → PM_3 → LL], [LL → PM_4 → PM_5 → LL], and [LL → PM_4 → PM_5 → PM_6 → LL], respectively. The processing parameters are as follows: κ_1 = 580 pcs, α_{11} = 145 s, β_{11} = 16 s, α_{12} = 348 s, β_{12} = 24 s; κ_2 = 600 pcs, α_{21} = 326 s, β_{21} = 28 s, α_{22} = 326 s, β_{22} = 28 s; κ_3 = 550 pcs, α_{31} = 156 s, β_{31} = 15 s, α_{32} = 139 s, β_{32} = 18 s, α_{33} = 145 s, β_{33} = 16 s; ρ = 6 s, θ = 2 s, π = 14 s.

For this example, the WFP is {(2, 1), ([1], [1]), ([1], [1],

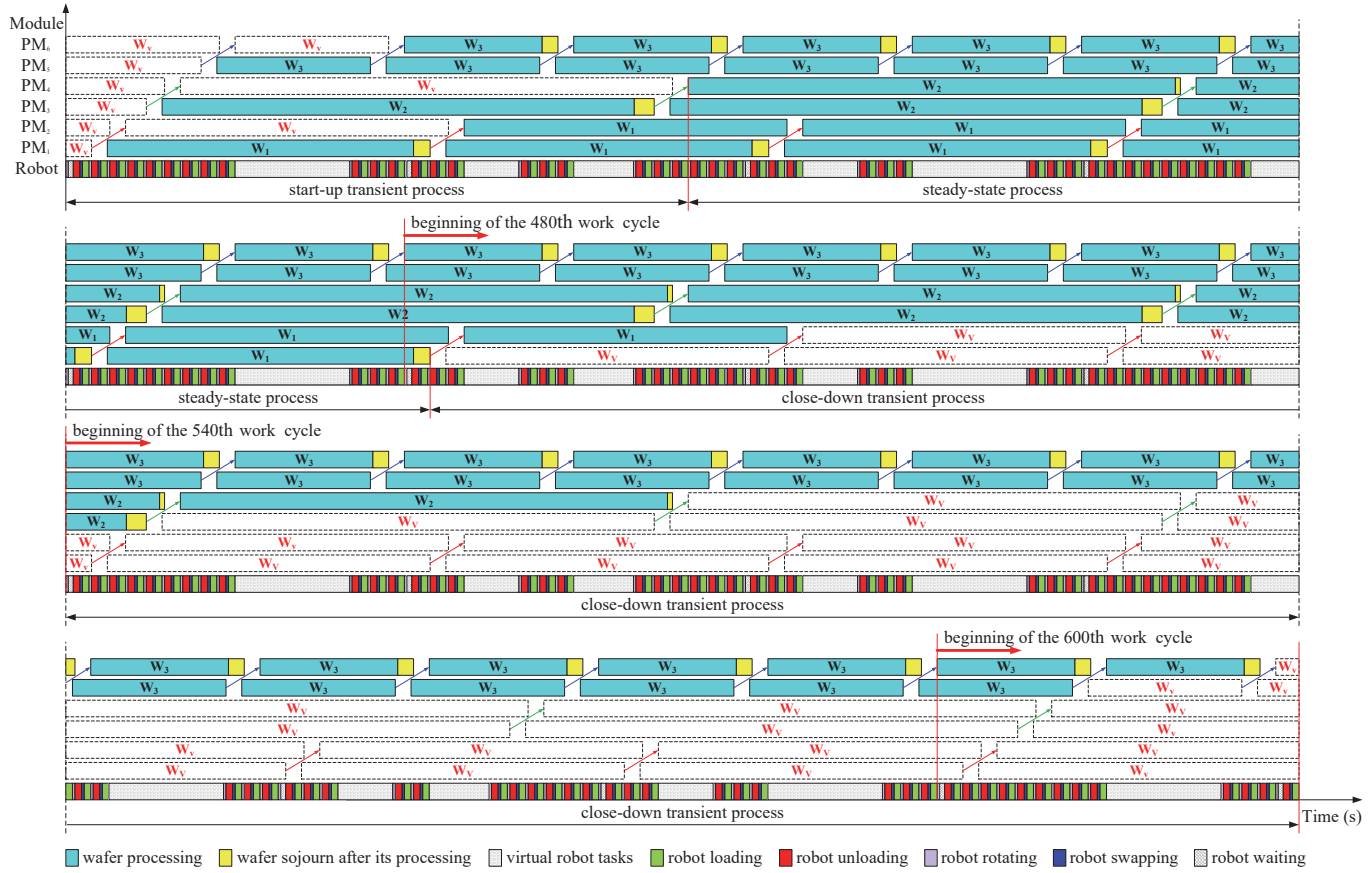


Fig. 4. Gantt chart of the obtained schedule for Example 1.

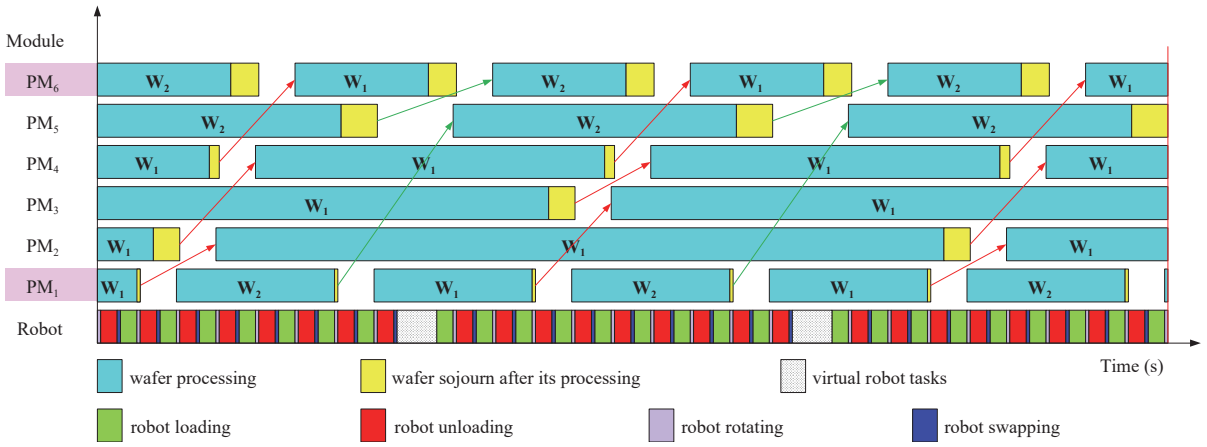


Fig. 5. Gantt chart of the obtained schedule for Example 2.

1)}. Then, $\psi_{R_c} = (3 + 4) \times (14 + 2) = 112$ s, $\eta_1 = 2$, and $\eta_2 = \eta_3 = 1$. Thus, we have $\varphi_{11\mathrm{L}} = 123$ s, $\varphi_{11\mathrm{U}} = 129$ s, $\varphi_{12\mathrm{L}} = 128$ s, $\varphi_{12\mathrm{U}} = 142$ s, $\varphi_{21\mathrm{L}} = 114$ s, $\varphi_{21\mathrm{U}} = 135$ s, $\varphi_{22\mathrm{L}} = 132$ s, $\varphi_{22\mathrm{U}} = 150$ s, $\varphi_{31\mathrm{L}} = 114$ s, $\varphi_{31\mathrm{U}} = 135$ s, $\varphi_{32\mathrm{L}} = 132$ s, $\varphi_{32\mathrm{U}} = 150$ s, $\varphi_{33\mathrm{L}} = 122$ s, and $\varphi_{33\mathrm{U}} = 130$ s. Then, $\bigcap_{\lambda\in\mathbb{N}_3,\mu\in\mathbb{N}_{n_\lambda}}[\varphi_{\lambda\mu\mathrm{L}}, \varphi_{\lambda\mu\mathrm{U}}] = \emptyset$, and $\varphi_{\mathrm{Lmax}} = 132$ s $> \psi_{R_c}$. Meanwhile, $\varphi_{11\mathrm{L}} < \varphi_{\mathrm{Lmax}}$ and $\varphi_{33\mathrm{L}} < \varphi_{\mathrm{Lmax}}$. However, $\varphi_{\mathrm{Lmax}} - (\varphi_{\mathrm{Lmax}} - \varphi_{11\mathrm{L}})m_{11}\zeta_{11}\eta_1 - (\varphi_{\mathrm{Lmax}} - \varphi_{33\mathrm{L}})m_{33}\zeta_{33}\eta_3 = 132 - (132 - 129) \times 2 \times 2 \times 1 - (132 - 130) \times 1 \times 2 \times 1 = 116$ s $> \psi_{R_c}$. Hence, the conditions given by Case 1 of Theorem 2 are satisfied. According to Algorithm 2, a feasible cyclic schedule can be obtained by setting $\psi = \psi_{\lambda\mu} = \psi_{\sigma R} = \psi_{R_c} = 132$ s, $\omega_{11} = (\varphi_{\mathrm{Lmax}} - \varphi_{11\mathrm{L}})m_{11}\zeta_{11}\eta_1 = 12$ s, $\omega_{33} = (\varphi_{\mathrm{Lmax}} - \varphi_{33\mathrm{L}})m_{11}\zeta_{11}\eta_1 = 4$ s, $\omega_2^{V_a} = 4$ s, $\omega_{10}^{P} = 4$ s, and the rest of robot waiting times are equal to zeros. The optimal quantity ratio of wafer types is $R_{\mathrm{opt}} = 1 : 1 : 1$. Thus, the quantities of the virtual wafers added necessarily are as follows: $\kappa_1^v = 20$ pcs, $\kappa_2^v = 0$ pcs, and $\kappa_3^v = 50$ pcs. Fig. 6 depicts the Gantt chart of the obtained schedule.

B. Schedule Implementation

As outlined in the previous sections, the cyclic schedule obtained in this paper is applicable to the steady state process

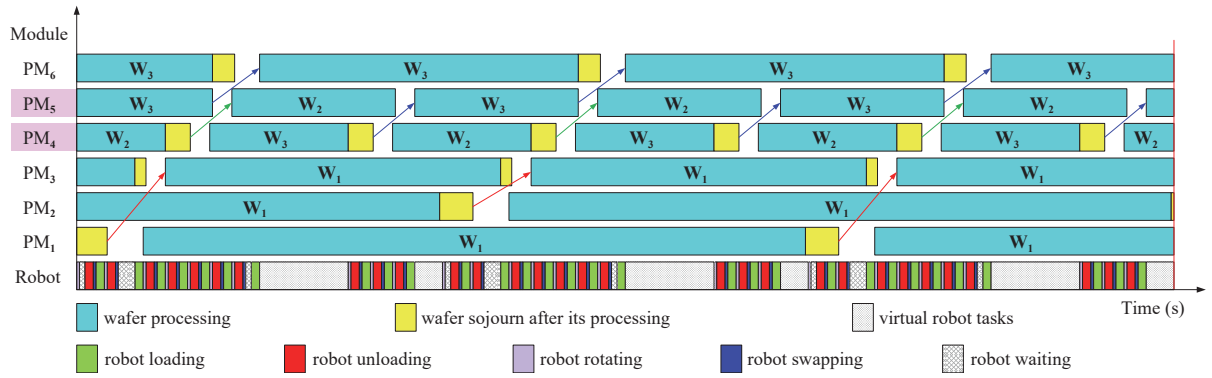


Fig. 6. Gantt chart of the obtained schedule for Example 3.

TABLE I
COMPARISON OF TYPICAL WORKS

Typical works	Tool type	Considered constraints			
		Quantity of wafer types (k)	Shared types	Parallel PMs	WRTCs
Lee *et al.* [54]	Single- and dual-arm cluster tools	$k = 2$	No	No	No
Lee *et al.* [55]	Single- and dual-arm cluster tools	$k = 2$	Yes	No	No
Ko *et al.* [56]	Dual-arm cluster tools	$k \geq 2$	No	Yes	No
Wang *et al.* [46]	Single-arm cluster tools	$k \geq 2$	Yes	Yes	Yes
This paper	Dual-arm cluster tools	$k \geq 2$	Yes	Yes	Yes

only. Before entering the steady state process, as known, the system will go through the start-up transient process that starts from the time the cluster tool begins to accept raw wafers to the point at which the full work cycle (i.e., the steady state) is reached. More crucially, the obtained steady state schedule demands distinctive requirements for the tool status while the cluster tool is going to the steady state. That is, the detailed status before the steady state must be compatible with that required by the obtained schedule. Thus, the robot activities during the start-up transient process should be accurately dispatched such that the system can converge to the desired state smoothly. When there is no longer raw wafers that are loaded into the tool, the system begins to go to the close-down transient process. Although the obtained schedule demands no special requirements for the tool status during the switching process, which is different from what the start-up transient process requires, we continue to hope that the close-down transient process can be sped up as soon as possible. Despite some studies with respect to the transient process scheduling for the single-wafer-type fabrication [48]–[53], none of them can be applied to the multi-type one. Thus, corresponding issues remain challenging and open. However, the primary goal pursued in this paper is to find the feasible cyclic steady state schedule and then implement it successfully. Inspired by the methods presented in [13], [14], we adopt the virtual wafer technology to avoid the transient state for implementing the obtained schedule.

We take Example 1 as an instance to illustrate how to avoid the transient state by using the virtual wafer technology. We first assume that the system has been in steady state and each PM is occupied by a virtual wafer (denoted by W_V) at the tool boot time. In both the start-up and close-down transient processes, the cluster tool is manipulated in accordance with the obtained steady state schedule. Specially, in the start-up transient process, the robot catches the real wafers and loads them into PMs, and the system will not reach the steady state until all the virtual wafers in PMs are replaced by the real ones. Such operation process is illustrated in Fig. 4, and it shows that the system begins to get into the steady state during the fourth work cycle. In the close-down transient process, the virtual wafers will be loaded into the tool as the supplement of the real ones. This manipulation will be continued until the entire real wafers complete their processing at the entire process steps. As shown in Fig. 4, the final wafer of the first (resp., second and third) type will complete its processing at the first/final process step during the 480th/482nd (resp., 540th/543rd and 600th/601st) work cycle. Note that the robot operations involving the virtual wafers aim at guaranteeing that the system keeps running according to the steady state schedule throughout the whole processes including the start-up transient, steady state, and close-down transient process. It is clear that this is easy to realize.

C. Comparison With Existing Works

Table I summarizes typical works with respect to the scheduling of multiple wafer types. In our earlier work [46], we investigate the scheduling problems of single-arm cluster tools with multiple wafer types and WRTCs. In this paper, we conduct these issues in dual-arm cluster tools. Compared with single-arm cluster tools, dual-arm ones present distinct characteristics in multiple aspects such as the robot operation strategy, temporal properties, and systematic schedulability, due to the widely different structures. Thus, compared with the work [46] for single-arm cluster tools, this paper for dual-

arm ones is not a typical epsilon difference work, even though both of them assume identical constraints. Compared with previous works [54]–[56], this paper achieves some progresses on the scheduling and control of dual-arm cluster tools with multiple wafer types being concurrently processed. First, WRTCs are not considered in [54]–[56], while they are taken into account in this paper. Second, the work in [54] and [55] merely address the concurrent processing of two wafer types, whereas [56] and ours can handle more than two types. Third, parallel PMs (i.e., series-parallel wafer flow patterns) are not considered in [54] and [55], but they are conducted in [56] and ours. However, the approach in [56] assumes that each wafer type has identical wafer flow patterns, whereas ours does not make such an assumption, as shown in Examples 2 and 3. Finally, for revealing the PM sharing behaviors, we investigate both non-shared and shared types via an integrated framework, as shown in Example 3, rather than conduct only one scenario, i.e., either non-shared types [54], [56] or shared types [55].

VI. Conclusion

This work focuses on the scheduling problems of dual-arm cluster tools with multiple wafer types and residency time constraints. A novel robot operation strategy, namely the multiplex swap sequence, is proposed to realize the stable mixed-model wafer fabrication. Based on this novel strategy, an easy-to-implement robot manipulation sequence for manifold flow patterns can be produced effortlessly. To balance the uneven workloads among process steps, a virtual module technology is developed. In addition to the presented necessary and sufficient conditions for schedulability testing, efficient algorithms with polynomial complexity are developed to find the cyclic schedule. The methodology presented in this paper can handle diverse complex multi-type wafer flow patterns. Meanwhile, its correctness, effectiveness, and adaptability are validated by illustrative examples.

More significantly, several orientations deserve further explorations. First, there is still room for improvement. For instance, how to find the optimal schedule when the actual quantitative proportion is not equal to R_{opt}? Especially, how to tackle with the scheduling problems when the shared types have different processing parameters at the shared steps? Second, it remains open and challenging for the scheduling and control problems with respect to the concurrent processing of multiple wafer types considering diverse operational requirements, e.g., revisiting processes, activity time variation, and cleaning operation. Third, it is of significance to investigate the concurrent processing of multiple wafer types within different tool architectures such as in-line cluster tools [4] and multi-cluster tools [2], [3]. In brief, these aforementioned topics deserve more attentiveness and we will investigate these issues in the future work.

References

[1] T.-S. Yu and T.-E. Lee, "Scheduling dual-armed cluster tools with chamber cleaning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 218–228, Jan. 2019.

[2] S. W. Ding, J. G. Yi, and M. T. Zhang, "Multicluster tools scheduling: an integrated event graph and network model approach," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 3, pp. 339–351, Aug. 2006.

[3] J. G. Yi, S. W. Ding, D. Z. Song, and M. T. Zhang, "Steady-state throughput and scheduling analysis of multi-cluster tools: a decomposition approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 2, pp. 321–336, Apr. 2008.

[4] H.-J. Kim, J.-H. Lee, S. Baik, and T.-E. Lee, "Scheduling in-line multiple cluster tools," *IEEE Trans. Semicond. Manuf*, vol. 28, no. 2, pp. 171–179, May 2015.

[5] L. Mönch, J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose, "A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations," *J. Sched.*, vol. 14, no. 6, pp. 583–599, Dec. 2011.

[6] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A steady state throughput analysis of cluster tools: dual-blade versus single-blade robots," *IEEE Trans. Semicond. Manuf.*, vol. 10, no. 4, pp. 418–424, Nov. 1997.

[7] R. S. Srinivasan, "Modeling and performance analysis of cluster tools using Petri nets," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 394–403, Aug. 1998.

[8] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, "Modeling and implementing a real-time scheduler for dual-armed cluster tools," *Comput. Ind.*, vol. 45, no. 1, pp. 13–27, May 2001.

[9] W. M. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Robot. Autom. Mag.*, vol. 17, no. 5, pp. 562–575, Oct. 2001.

[10] S. Rostami, B. Hamidzadeh, and D. Camporese, "An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 609–618, Oct. 2001.

[11] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, "Scheduling analysis of time-constrained dual-armed cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 3, pp. 521–534, Aug. 2003.

[12] T.-E. Lee and S.-H. Park, "An extended event graph with negative places and tokens for time window constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 4, pp. 319–332, Oct. 2005.

[13] N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, "A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 2, pp. 224–237, May 2008.

[14] N. Q. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 303–315, Apr. 2010.

[15] U. Wikborg and T.-E. Lee, "Noncyclic scheduling for timed discreteevent systems with application to single-armed cluster tools using Pareto-optimal optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 699–710, Jul. 2013.

[16] J.-H. Kim and T.-E. Lee, "Schedulability analysis of timed-constrained cluster tools with bounded time variation by an extended Petri net," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 490–503, Jul. 2008.

[17] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Real-time scheduling of single-arm cluster tools subject to residency time constraints and bounded activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 564–577, Jul. 2012.

[18] N. Q. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.

[19] C. R. Pan, Y. Qiao, N. Q. Wu, and M. C. Zhou, "A novel algorithm for wafer sojourn time analysis of single-arm cluster tools with wafer residency time constraints and activity time variation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 5, pp. 805–818, May 2015.

[20] H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Schedulability analysis for noncyclic operation of time-constrained cluster tools with time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1409–1414, Jul. 2016.

[21] H.-Y. Lee and T.-E. Lee, "Scheduling single-arm cluster tools with reentrant wafer flows," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 226–240, May 2006.

[22] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net-based

scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 42–55, Jan. 2011.

[23] N. Q. Wu, M. C. Zhou, F. Chu, and C. B. Chu, "A Petri-net-based scheduling strategy for dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 43, no. 5, pp. 1182–1194, Sep. 2013.

[24] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 286–300, Feb. 2014.

[25] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Schedulability and scheduling analysis of dual-arm cluster tools with wafer revisiting and residency time constraints based on a novel schedule," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 45, no. 3, pp. 472–484, Mar. 2015.

[26] F. J. Yang, N. Q. Wu, Y. Qiao, M. C. Zhou, and Z. W. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[27] Y. Qiao, N. Q. Wu, F. J. Yang, M. C. Zhou, Q. H. Zhu, and T. Qu, "Robust scheduling of time-constrained dual-arm cluster tools with wafer revisiting and activity time disturbance," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 49, no. 6, pp. 1228–1240, Jun. 2019.

[28] Y. Qiao, N. Q. Wu, C. R. Pan, and M. C. Zhou, "How to respond to process module failure in residency time-constrained single-arm cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 27, no. 4, pp. 462–474, Nov. 2014.

[29] Y. Qiao, C. R. Pan, N. Q. Wu, and M. C. Zhou, "Response policies to process module failure in single-arm cluster tools subject to wafer residency time constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 1125–1139, Jul. 2015.

[30] H. Kim, H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Scheduling dual-armed cluster tools with cleaning processes," *Int. J. Prod. Res.*, vol. 51, no. 12, pp. 3671–3687, Dec. 2013.

[31] T.-S. Yu, H.-J. Kim, and T.-E. Lee, "Scheduling single-armed cluster tools with chamber cleaning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 705–716, Apr. 2018.

[32] F. J. Yang, N. Q. Wu, K. Z. Gao, C. J. Zhang, Y. T. Zhu, R. Su, and Y. Qiao, "Efficient approach to cyclic scheduling of single-arm cluster tools with chamber cleaning operations and wafer residency time constraint," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 196–205, May 2018.

[33] W. K. V. Chan, S. W. Ding, J. G. Yi, and D. Z. Song, "Optimal scheduling of multicluster tools with constant robot moving times, Part II: tree-like topology configurations," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 17–28, Jan. 2011.

[34] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri net-based optimal one-wafer scheduling of single-arm multi-cluster tools in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 4, pp. 578–591, Nov. 2013.

[35] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri net-based polynomially complex approach to optimal one-wafer cyclic scheduling of hybrid multi-cluster tools in semiconductor manufacturing," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 44, no. 12, pp. 1598–1610, Dec. 2014.

[36] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 117–125, Feb. 2015.

[37] L. P. Bai, N. Q. Wu, Z. W. Li, and M. C. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

[38] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Optimal one-wafer cyclic scheduling of time-constrained hybrid multicluster tools via Petri nets," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 47, no. 11, pp. 2920–2932, Nov. 2017.

[39] F. J. Yang, N. Q. Wu, Y. Qiao, and R. Su, "Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018.

[40] F. J. Yang, N. Q. Wu, Y. Qiao, and R. Su, "Optimal one-wafer cyclic scheduling of hybrid multirobot cluster tools with tree topology," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst*, vol. 48, no. 2, pp. 289–298, Feb. 2018.

[41] T. Nishi, Y. Watanabe, and M. Sakai, "An efficient deadlock prevention policy for noncyclic scheduling of multicluster tools," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1677–1691, Oct. 2018.

[42] Q. H. Zhu, Y. Qiao, and N. Q. Wu, "Optimal integrated schedule of entire process of dual-blade multi-cluster tools from start-up to closedown," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 553–565, Mar. 2019.

[43] C. R. Pan, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Scheduling cluster tools in semiconductor manufacturing: recent advances and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 586–601, Apr. 2018.

[44] Y. Lim, T.-S. Yu, and T.-E. Lee, "A new class of sequences without interferences for cluster tools with tight wafer delay constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 392–405, Jan. 2019.

[45] F. J. Yang, N. Q. Wu, Y. Qiao, M. C. Zhou, R. Su, and T. Qu, "Modeling and optimal cyclic scheduling of time-constrained singlerobot- arm cluster tools via Petri nets and linear programming," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 50, no. 3, pp. 871–883, Mar. 2020.

[46] J. P. Wang, C. R. Pan, H. S. Hu, L. Li, and Y. Zhou, "A cyclic scheduling approach to single-arm cluster tools with multiple wafer types and residency time constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1373–1386, Jul. 2019.

[47] J.-H. Lee, H.-J. Kim, and T.-E. Lee, "Scheduling lot switching operations for cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 4, pp. 592–601, Nov. 2013.

[48] T.-K. Kim, C. Jung, and T.-E. Lee, "Scheduling start-up and close-down periods of dual-armed cluster tools with wafer delay regulation," *Int. J. Prod. Res.*, vol. 50, no. 10, pp. 2785–2795, May 2012.

[49] C. R. Pan, Y. Qiao, M. C. Zhou, and N. Q. Wu, "Scheduling and analysis of start-up transient processes for dual-armed cluster tools with wafer revisiting," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 2, pp. 160–170, May 2015.

[50] D.-K. Kim, T.-E. Lee, and H.-J. Kim, "Optimal scheduling of transient cycles for single-armed cluster tools with parallel chambers," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 1165–1175, Apr. 2016.

[51] Y. Qiao, M. C. Zhou, N. Q. Wu, and Q. H. Zhu, "Scheduling and control of startup process for single-arm cluster tools with residency time constraints," *IEEE Trans. Control. Syst. Technol.*, vol. 25, no. 4, pp. 1243–1256, Jul. 2017.

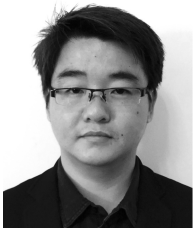
[52] Q. H. Zhu, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Scheduling transient processes for time-constrained single-arm robotic multi-cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 3, pp. 261–269, Aug. 2017.

[53] Q. H. Zhu, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.

[54] J.-H. Lee, H.-J. Kim, and T.-E. Lee, "Scheduling cluster tools for concurrent processing of two wafer types," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 525–536, Apr. 2014.

[55] J.-H. Lee, H.-J. Kim, and T.-E. Lee, "Scheduling cluster tools for concurrent processing of two wafer types with PM sharing," *Int. J. Prod. Res.*, vol. 53, no. 19, pp. 6007–6022, Oct. 2015.

[56] S.-G. Ko, T.-S. Yu, and T.-E. Lee, "Scheduling dual-armed cluster tools for concurrent processing of multiple wafer types with identical job flows," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1058–1070, Jul. 2019.

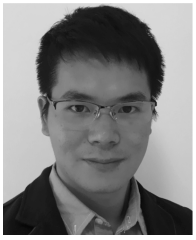


Jipeng Wang (S'17) received the B.S. and M.S. degrees in mechanical engineering from Hubei University of Technology, in 2011 and Jiangxi University of Science and Technology, in 2016, respectively. He is currently pursuing the Ph.D. degree in control theory and control engineering at the School of Mechano-Electronic Engineering, Xidian University, His research interests include Petri nets, discrete event systems, and production planning.

Hesuan Hu (M'11–SM'12) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in electromechanical engineering from Xidian University, in 2003, 2005, and 2010, respectively. He is currently a Full Professor with Xidian University. His research interests include discrete event systems and their supervisory control techniques, Petri nets, automated manufacturing systems, multimedia streaming systems, and artificial intelligence. In the aforementioned areas, he has over 120 publications in journals, book chapters, and conference proceedings. Dr. Hu was a recipient of many national and international awards, including the Franklin V. Taylor Outstanding Paper Award from the IEEE Systems, Man, and Cybernetics Society in 2010, and the finalist of the Best Automation Paper from the IEEE RAS Society in 2013, 2016, and 2017. He is an Associate Editor for *IEEE Control Systems Magazine*, *IEEE Robotics and Automation Magazine*, *IEEE Transactions on Automation Science and Engineering*, *Journal of Intelligent Manufacturing*, etc. He was on the Editorial Board of over ten international journals.

Chunrong Pan (M'14–SM'19) received the M.S. degree in mechatronics engineering from Shantou University, and the Ph.D. degree in mechanical engineering from Guangdong University of Technology, in 2006 and 2010, respectively. From 1997 to 2011, he was with Shantou University. He joined Jiangxi University of Science and Technology, in 2011, where he is currently a Full Professor with the Department of Mechatronics Engineering. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2013 to 2014, and the Bournemouth University, Poole, Dorset, UK, from 2018 to 2019, respectively. He has published over ten research papers in international journals and conferences. His research interests include modeling, simulation, scheduling, and control of integrated manufacturing equipment.

Yuan Zhou received the M.S. degree in computational mathematics from Zhejiang Sci-Tech University, in 2015. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published several research papers in various journals and conferences, such as the *IEEE Transactions on Fuzzy Systems*, *IEEE Transactions on Systems*, *Man*, *and Cybernetics: Systems*, *IEEE Transactions on Automation Science and Engineering*, the *IEEE Transactions On Reliability*, the *IEEE International Conference on Automation Science and Engineering*, and the *International Conference on Software Engineering*. His research interests include robot motion control and planning, multirobot systems, discrete event systems, Petri nets, and system modeling.



Liang Li received the M.S. degree in mechanical engineering from Jiangxi University of Science and Technology, in 2015. He is currently pursuing the Ph.D. degree in control theory and control engineering at the School of Mechano-Electronic Engineering, Xidian University. He is also pursuing the Ph.D. degree under a joint international supervision agreement with the Department of Computer Engineering, Electrical Engineering and Applied Mathematics, University of Salerno, Fisciano, Italy. His research interests include Petri nets, control and optimization of timed discrete event systems.